

HAKING

PRACTICAL PROTECTION

IT SECURITY MAGAZINE

VOL.15, NO. 02

The background of the cover is a stylized illustration of a city skyline with numerous skyscrapers. Overlaid on this is a large number of drones, depicted in various sizes and orientations, flying across the sky. The drones are primarily black with blue propellers. The overall color palette is dominated by purples, blues, and yellows.

DRONE HACKING

EXPLOITATION AND VULNERABILITIES

WHAT IF A DRONE IS HACKED?

DEFENSE TECHNIQUES AGAINST ATTACKS ON UAV

FINDING SECURITY VULNERABILITIES IN UAV

AND MORE...

HAKING

TEAM

Editor-in-Chief

Joanna Kretowicz

joanna.kretowicz@eforensicsmag.com

Editors:

Marta Sienicka

sienicka.marta@haking.com

Dominika Zdrodowska

dominika.zdrodowska@eforensicsmag.com

Marta Strzelec

marta.strzelec@eforensicsmag.com

Bartek Adach

bartek.adach@pentestmag.com

Proofreader:

Lee McKenzie

Senior Consultant/Publisher:

Paweł Marciniak

CEO:

Joanna Kretowicz

joanna.kretowicz@eforensicsmag.com

Marketing Director:

Joanna Kretowicz

joanna.kretowicz@eforensicsmag.com

DTP

Marta Sienicka

sienicka.marta@haking.com

Cover Design

Hiep Nguyen Duc

Joanna Kretowicz

Publisher

Haking Media Sp. z o.o.

02-676 Warszawa

ul. Bielawska 6/19

Phone: 1 917 338 3631

www.haking.org

BETATESTERS &

PROOFREADERS

Lee McKenzie

Hammad Arshed

Ali Abdollahi

Robert Fling

Paul Mellen

Bernhard Waldecker

Avi Benchimol

Amit Chugh

Kevin Goosie

Raymond Obinaju

Tom Updegrove

All trademarks, trade names, or logos mentioned or used are the property of their respective owners.

The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

Dear readers,

Drones are a growing threat to law enforcement and security specialists. Low-cost and easy to use, drones can carry out surveillance, capture data, or disrupt networks. Making matters worse, drones are hard to detect and defeat. Their growing popularity is proportional to the number of exploits found in UAVs by hackers. In this edition, we would like to focus on the strong and weak points of drones. What are their vulnerabilities, how to exploit them? On the other hand, you will see what steps to take to secure your UAV.

Let's see what's inside!

The first part of this edition is focused on vulnerabilities and exploits that can be found in a UAV. In the article *Exploring Security Vulnerabilities of UAV*, you will learn how attacks such as Man-in-the-Middle can affect your device (there are other techniques presented as well). How to find *Security Vulnerabilities in Unmanned Aerial Vehicles Using Software* is another article that will dive into weak points of drones, and help you understand how a UAV can be attacked. Following that, *Cyber Attack Vulnerabilities Analysis for UAV* is focused on the methods used by hackers while performing an attack. After the analysis, the authors looked closer at the post-attack behavior of the autopilot system through simulation.

Now that we know where to find vulnerabilities and how to exploit them, it's time to learn more about securing your equipment.

We start with a review of the current situation in *Counter UAV strategies*. You will learn what are the most important prevention techniques used by specialists to secure drones. Moving forward, *Protecting the UAV from Cyber Attacks* and *Defense Techniques Against Cyber Attacks on UAV* are mainly focused on the best techniques against cyber attacks on drones including wireless network encryption and intrusion detection system.

But that's not all! We recommend the article *Security Analysis of FHSS-type Drone Controller* that presents an investigation of security in drone controllers. It's a different approach, surprisingly rarely talked about, so we hope that you will find it interesting. *Wireless Communications with UAV* will show you, by introducing basic networking architecture and main channel characteristics, how to efficiently use wifi communication to make your drone more effective while performing various tasks. *Integration of Machine Learning to simplify the Analysis in Security Operations Center (SOC)* will close this edition.

We hope that no matter where you are, you are safe, taking care of yourself and your loved ones. We are all coping with COVID-19 in different ways, and facing different challenges. Those are difficult times, but together, we are strong. Stay safe, stay focused, and don't give up.

Enjoy the reading,

Hakin9 Editorial Team

Contents



Are drones safe from Humans – What if a drone is hacked!

by Nazneen Khan



Exploring Security Vulnerabilities of Unmanned Aerial Vehicles

by Nils Miro Rodday
Ricardo de O. Schmidt
Aiko Pras



Finding Security Vulnerabilities in Unmanned Aerial Vehicles Using Software Verification

by Omar M. Alhawi
Mustafa A. Mustafa
Lucas C. Cordiro



Cyber Attack Vulnerabilities Analysis for UAV

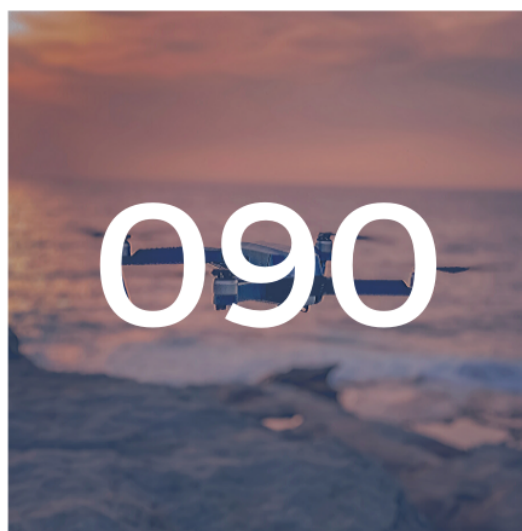
by Brandon Wampler
James Goppert
Inseok Hwang

Contents



Counter UAV strategies – A current day review

by Alan Roder



Protecting the Unmanned Aerial Vehicle from Cyberattacks

by Jesus Nunez
Vincent Tran
Ajay Katangur



Defense Techniques Against Cyber Attacks on UAV

by Charan Gudla
Md. Shohel Rana
Andrew H. Sung



Security Analysis of FHSS-type Drone Controller

by Kibum Choi
Youngseok Park

Contents



Wireless Communications with UAV: Opportunities and Challenges

by Yong Zeng
Rui Zhang
Teng Joon Lim



Integration of Machine Learning to simplify the Analysis in Security Operations Center (SOC)

by Chirath De Alwis



Are drones safe from Humans – What if a drone is hacked!

Nazneen Khan



ABOUT THE AUTHOR

NAZNEEN KHAN

Keen in working with new technologies like ROS, computer vision and doing freelancing in fields of robotics, automation, drones, machine vision, artificial intelligence and deep learning.

When we talk about drones the first thing that comes to our mind is a UAV (Unmanned Aerial Vehicle) with a camera which can fly and give us live recording of an event or which can be used to click high definition pictures or videos for tourism of lakes or waterfalls, but there is more to it.

Nowadays, drones found interest in not only entertainment industry but also in defense, transportation, logistics, agriculture and many more. In 2017, the drone company called Griff Aviation launched Griff 300 model which can carry around 550 lbs (226kg) of weight and drone manufacturers are also planning to deploy drones as human transportation vehicle son. And as technology advances the flight time of drone is also increased and the longest flight time drone was developed called US-1 that can fly a whopping two hours on a single charge. Now, companies like amazon are planning to deploy to drones to deliver packages or food products thereby making the package available for attackers to target it and steal the package by simply playing with its control signal or GPS, till the time operator comes to know that the package is stolen it will be too late and package will be gone.

Payload is the most crucial part of the drone as it decides the purpose of flight and if the drone is used for transportation of goods or even in defense industries for transporting guns, missiles from one place to other it becomes more important to protect it. Defense drones with guns suspended on a gimbal control to point the target can be used to kill people in wars or terror attacks and as such till now there are no federal laws that make putting gun on your drone inherently illegal, according to the former head of Federal Aviation Administration's unmanned aircraft office. Many countries and groups already use small military drones that can drop grenades or fly into a target to detonate an explosive.

Now autonomous drones based on latest technologies like SLAM (Simultaneous localization and mapping) and LiDAR based localization with the help of single board computers can create a 3D map of its surrounding autonomously and sometime through the wall using Wi-Fi and 3D cameras without the use of pilot or remote-control operator and as these data are very crucial and if 3D map of any bank or government offices reaches terrorist or thieves this can lead to dangerous consequences. Drones are now being used to hack servers and also to spy on networks people using Wi-Fi and 3D technologies drones are now capable of watching through wall, so drone manufacturing companies need to take some serious measures to protect the drone from being hacked by various measures. The FAA claims that as long as drones don't endanger people, drones can legally hover just above private property. Currently, the FAA forbids flying hobby drones over 400 feet, and drones may not interfere with official or emergency response aircraft engaged in public safety operations like firefighting.

When we talk about cybersecurity various actions have already been taken by the government to take legal action against any cyberattacks and there are cyberlaws that governs the digital dissemination of both information and software, information security and electronic commerce. But the problem that comes with cyberlaws with drones is that, drones are mainly operated by remote control and it is very difficult to track the location of the remote pilot by the law of enforcement authorities, for which no such regulatory rules or regulations are present specifically. Also, there are no such laws to penalize activities like invading persons privacy by taking videos or pictures of people or their private properties with drones.

Now, imagine drone swarms (multiple unmanned platforms and/or weapons deployed to accomplish a shared objective, with the platforms and/or weapons autonomously altering their behavior based on communication with one another) being hacked all together can lead to massive destruction like terror attacks or war. So, these were few highlights on where we stand now on cyberlaws and drone technologies.

Now let's talk about how a drone is hacked:

The most important step in hacking a drone is getting access to the position of the drone or locating it which can be done by using radio frequency sensors, radar, infrared or acoustic sensors. As the hackers knows where drone is located then he can use several different ways to hack it:

- a. **Simulating the GPS signal** – A GPS simulator with RF front end can generates GPS signals of higher strengths than the coming from GPS satellites and feeding these signals to the GPS receiver of the drone and thereby drone can be directed to fly to any false location where hacker wants and can lead to various accidents like crashing or the drone can be directed to a location in proximity with hacker so that it can be stolen. GPS jammers can also hinder GPS receiver from working properly
- b. **Hijacking uplink/ download signal** – The operator/pilot communicates with the drone using commands and control signal and protocols like MAVLink and hacker can hijack these commands and cause similar effects like GPS jamming. However, few drones are equipped with return to home failsafe if they lose control signal. The radio signals are not generally encrypted and can be easily hacked using packet sniffer. But getting control over the commands gives the hacker full access to the drone system including its camera, gimbal and sensors which is very dangerous.
- c. **Using ADS-B system** – These system is used by Federal Aviation Administration to avoid air traffic and collision between drones and hackers can broadcast these false ADS-B signals from a hacked drone and cause confusion and fatal consequences.

Security measures taken by drone manufacturers to avoid hacking:

1. **Geofencing** - It is a virtual fence commonly known as perimeter, in which the manufacturer can create a limit to place where drone can fly by using GPS and also gets its current location and then can trigger alarm or return to home system if the drone breaches geofence. The best thing about geofencing is that it does not require anything physical and all work is done digitally by coding and can be deployed anywhere outdoor like farm, airport or lakes and it uses Google and Apple databases which saves our mapping work. The minimum size of geofence ranges from 20 to 50 metres but with advances in technologies you can go to minimum 2 metres as well. Drone companies nowadays also include geofencing as part of their toolbox such as Solo Custom geofence where the users can define its own four points on a map within their app thereby creating a defined area where drone is allowed to fly. However, if the drone gets hacked the hacker will do the first thing of removing geofence and altitude limit set by the manufacturers.

2. **Data encryption** - Logs are the most crucial data for any UAV pilots or manufacturers as it gives information related to the places where drone flew along with the altitude, velocity and also the payload information. These logs are transferred by drone to the ground station where pilot monitors data for breaches and also to check its performance. If the drone is used for surveillance purpose and the hackers attacks it, they will get all these valuable data which is dangerous. So, manufacturers have started using various hardware security measures like encrypting the drone logs and bundling them up with a unique key and using ciphering algorithms and various other data encryption technologies for protecting the logs and also avoiding attacker to hack the logs.
3. **Accompanying single board computers** - When we say companion computer it can be raspberry pi, Jetson Nano or any single board computer communicating with the flight controller using MAVLink protocol and any interface preferably serial. There are both pros and cons of adding additional computer to the UAV but can help to make the drone more secure by alarming the GCS if any problems have been encountered while communicating with the flight controller like Pixhawk and also by sending email or alarm to the pilot and thereby protecting the drone from attacks. But as said everything has drawbacks this one too like what if the hackers attack the single board computer, the manufacturers should take into account all these possibilities while testing the drone before deployment. By using proper encryption and disabling SSH (Secure Shell Remote login) attacks can be prevented.
4. **Other measures that can be taken** – The firmware of the drone should always be up to date. And use of strong password to protect the ground station app should be emphasized and also there should be a limit to the number of devices that are allowed to be connected to the access points. Additionally, remote storage provider should be secured with two factor authentication and full encryption. Operator should try to use VPN to secure the digital connection from their laptop with anti-virus protection to the storage server to protect their drone login credentials as well as data extraction.

There is no doubt that as technology advances the measures adopted by the government or manufacturers to improve drone safety will be taken into account more and there will be more rules and regulations for them in addition to ones already present. But, not to forget with advancements the hackers will also come with new ways to attack the drone and cause dreadful consequences. So, to conclude the question is not how secure are drones from attackers but it is - How secure are we humans from a hacked drone!



Exploring Security Vulnerabilities of Unmanned Aerial Vehicles

Nils Miro Rodday

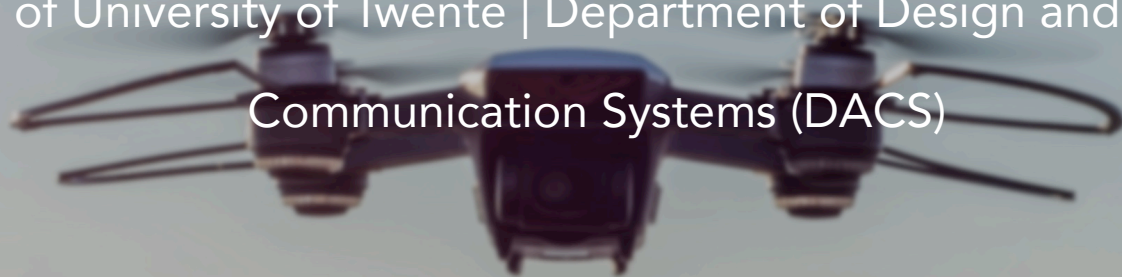
Ricardo de O. Schmidt

Aiko Pras

ABOUT THE AUTHOR

NILS MIRO RODDAY

Student of University of Twente | Department of Design and Analysis of
Communication Systems (DACS)





ABOUT THE AUTHOR

RICARDO DE O. SCHMIDT

I am a professor at the Computer Science Department of the University of Passo Fundo. I am actively contributing with researchers from the Information Sciences Institute of the University of Southern California. In 2014 I received a PhD degree from the University of Twente for my thesis Measurement-Based Link Dimensioning for the Future Internet.





ABOUT THE AUTHOR

AIKO PRAS

Aiko Pras is full professor in the area of Network Operations and Management with a focus on Internet Security at the Faculty of Electrical Engineering, Mathematics and Computer Science of the University of Twente, the Netherlands. He is member of the Design and Analysis of Communication Systems Group (DACS). In 1995 he received a Ph.D. degree from the same university for his thesis titled "Network Management Architectures" and in 2013 he was appointed as full professor. In 2016 he has been honoured with the IFIP/IEEE "Salah Aidarous Memorial Award" for providing unremitting service and dedication to the IT and Telecommunications Network Operations and Management community.

We are currently observing a significant increase in the popularity of Unmanned Aerial Vehicles (UAVs), popularly also known by their generic term, drones. This is not only the case for recreational UAVs, that one can acquire for a few hundred dollars, but also for more sophisticated ones, namely, professional UAVs, where the cost can reach several thousands of dollars. These professional UAVs are known to be largely employed in sensitive missions such as monitoring of critical infrastructures and operations by the police force. Given these applications, and in contrast to what we have been seeing for the case of recreational UAVs, one might assume that professional UAVs are strongly resilient to security threats. In this demo, we prove such an assumption wrong by presenting the security gaps of a professional UAV that is used for critical operations by police forces around the world. We demonstrate how one can exploit the identified security vulnerabilities, perform a Man-in-the-Middle attack, and inject control commands to interact with the compromised UAV. In addition, we discuss appropriate countermeasures to help improve the security and resilience of professional UAVs.

I. INTRODUCTION

The fact that recreational Unmanned Aerial Vehicles (UAVs), accessible to the general public, are not secure is not new. Several papers and news articles have been published showing that one can easily hack into these devices. However, recreational UAVs are hardly ever used for situations out of the leisure context. Examples of potential applications for professional UAVs are surveillance, border control and search & rescue. For sensitive and critical operations, professional UAVs are able to deliver the performance and functionalities that are needed. Examples of advanced features of these professional UAVs are long endurance and the ability to carry a heavy payload. Clearly, the more advanced the more expensive the device is, and professional UAVs can easily cost several thousands of dollars.

Given the range of applicability, one should expect that security is a top priority for professional UAVs. From our analysis on a professional UAV, kindly borrowed from its manufacturer (names and models are not disclosed due to a non-disclosure agreement), we have learned that this is not the case. This puts critical operations for which these UAVs are used in danger of failure at the very least.



Fig. 1. Architecture

To the best of our knowledge, there is no work in the literature that openly addresses the security issues of professional UAVs. In this demo, we show that professional UAVs are not as secure as one might expect. We demonstrate that by

learning how the UAV communicates with the remote controller, one can perform a Man-in-the-Middle (MitM) attack and potentially take control over the UAV, even at a distance of several kilometers from the actual UAV's controller. With our findings, we raise awareness within (i) the general public that use and trust such professional UAVs, (ii) the scientific community by showing that further research is needed in this area, and (iii) the manufacturers by showing the importance of implementing a higher level of security in their devices.

II. EXPLORING SECURITY FLAWS OF THE UAV

This research has been performed with a professional UAV, kindly borrowed from a high-end manufacturer. However, as the same hardware and software components are also used by other manufacturers, our approach and results can be extended to their respective UAVs.

A. System Architecture

Figure 1 shows that the UAV has telemetry, manual remote control (RC) and video links. The manual RC link uses common 2.4Ghz Graupner equipment and allows basic steering functionality within a range of 100m. The telemetry link allows for more advanced control features, such as setting way-points and automated flying. This is done from a tablet running the flight planning software, which is connected via WiFi 802.11 to the telemetry box that in turn forwards the communication to the UAV by using XBee 868LP chips. The telemetry link can reach with a range of several kilometers, much further than the manual RC link, allowing for full control of the UAV even if the UAV is far out of sight.

B. Security Vulnerabilities

We focused on the telemetry link of the UAV because of the broader range and the wider spectrum of control over the UAV. The telemetry link consists of two separate communication links, whereby the communication is forwarded between WiFi 802.11 and XBee 868LP by the telemetry box.

To communicate with an XBee 868LP chip the attacker needs to know the following connection parameters: PAN ID (network ID), BAUD rate, channel, destination high (DH) address and destination low (DL) address. PAN ID, BAUD rate and channel are all set to default values for every UAV and, hence, they can be considered of general knowledge. This way, a potential attacker still misses the DH and DL information. Theoretically, there are $18 \cdot 10^{18}$ possible combinations. However, the XBee 868LP chips respond to broadcast packets sent within their network, and this can be done through API-mode. The acknowledgement message for the broadcast contains the address of the sender, hence, revealing all available devices within the network.

Moreover, we have also identified a security gap in the WiFi link. The access point uses WEP (Wired Equivalent Privacy) as an encryption scheme and can, therefore, be cracked. An attack on the WiFi link can be performed as follows: (1) crack the password, (2) disconnect the original user, and (3) connect the attacker's tablet to the UAV.

However, to do so, the attacker must be within the range of the WiFi link (100m), which likely makes such an attack too risky.

C. Man-in-the-Middle Attack

As shown in Figure 2, in order to perform a Man-in-the-Middle attack on the XBee 868LP chips we need to use “Remote AT Commands”. This feature allows for the attacker to remotely change internal parameters of the XBee chips, such as DH and DL, and therefore reroute any traffic. The write command persists changes within memory, allowing for two different attack modes: temporary or persistent.

We have reversed-engineered both the flight computer and flight planning software. We were able to match commands transferred through the telemetry channel with specific functions within the UAV system. This enables an attacker to understand and alter existing packets in a meaningful way, or inject new packets to communicate with the flight computer.

D. Countermeasures

Fixing the security gaps we have identified within the studied UAV is not a straightforward procedure. First, secure encryption schemes should be used for the WiFi 802.11 access point connecting the tablet with the telemetry box. Second, data transferred through the XBee 868LP chip should not be sent in clear-text. Encryption should be used throughout the whole communication path. Three solutions could be employed: (1) XBee 868LP on-board encryption, which is the only solution that also mitigates the risk of Remote AT Commands, (2) dedicated hardware encryption in case the throughput drops significantly with XBee 868LP encryption, and (3) application layer encryption.

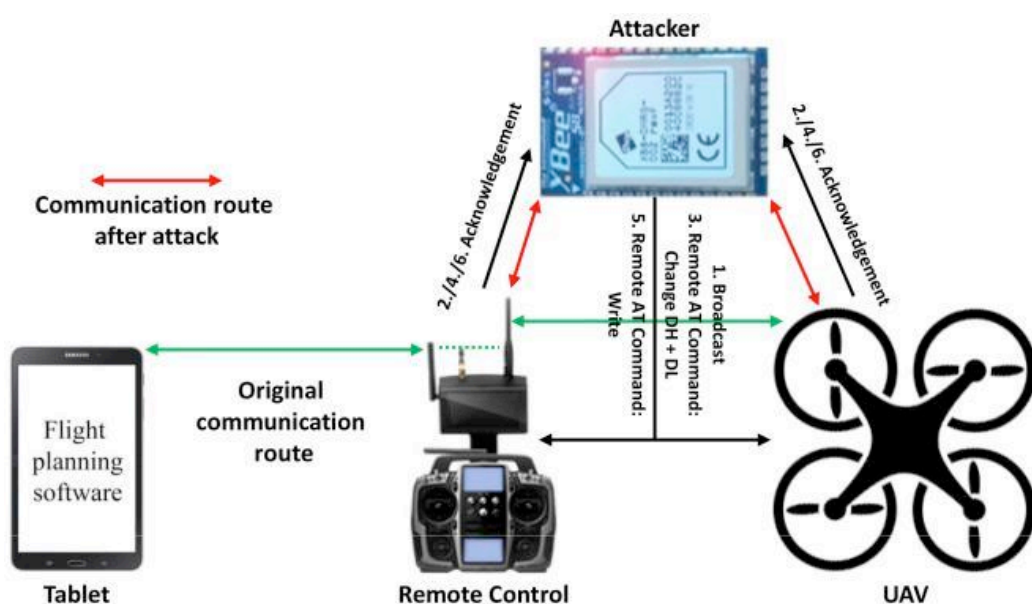


Fig. 2. Man-in-the-Middle attack

III. FINAL CONSIDERATIONS

In this article, we have shown that expensive professional UAVs can be hacked due to severe security vulnerabilities in their setup. We have shown that it is possible to effectively perform a Man-in-the-Middle attack from kilometers away by rerouting the traffic of the telemetry channel. Moreover, we have shown that by reverse-engineering the software involved in the communication of the UAV system, we can inject packets and control the UAV. Countermeasures to close these security vulnerabilities exist, but require the manufacturer to develop the system further and to patch every item sold so far.



Finding Security Vulnerabilities in Unmanned Aerial Vehicles Using Software Verification

Omar M. Alhawi

Mustafa A. Mustafa

Lucas C. Cordiro



ABOUT THE AUTHOR

OMAR M. ALHAWI

Department of Computer Science

School of Engineering



ABOUT THE AUTHOR

MUSTAFA A. MUSTAFA

Mustafa A. Mustafa is currently a Dame Kathleen Ollerenshaw Research Fellow in the Department of Computer Science at The University of Manchester.

He received the B.Sc. degree in communications from the Technical University of Varna, Varna, Bulgaria, in 2007, the M.Sc. degree in communications and signal processing from Newcastle University, Newcastle upon Tyne, U.K., in 2010, and the Ph.D. degree in computer science from The University of Manchester, Manchester, U.K., in 2015. He then was a post-doctoral research fellow with the imec-COSIC research group, Department of Electrical Engineering (ESAT), KU Leuven, Belgium. Since July 2018 he is a Dame Kathleen Ollerenshaw Research Fellow in the Department of Computer Science at The University of Manchester.



ABOUT THE AUTHOR

LUCAS C. CORDIRO

Lucas C. Cordeiro is a Senior Lecturer (Associate Professor) in the Department of Computer Science at the University of Manchester (UK), where he leads the Systems and Software Verification laboratory. He is also a collaborator in the Postgraduate Program in Electrical Engineering and Informatics at the Federal University of Amazonas (Brazil). He has co-authored more than 100 publications in the area of software model checking, automated testing, program synthesis and embedded & cyber-physical systems

The proliferation of Unmanned Aerial Vehicles (UAVs) embedded with vulnerable monolithic software has recently raised serious concerns about their security due to concurrency aspects and fragile communication links. However, verifying security in UAV software based on traditional testing remains an open challenge mainly due to scalability and deployment issues. Here we investigate software verification techniques to detect security vulnerabilities in typical UAVs. In particular, we investigate existing software analyzers and verifiers, which implement fuzzing and bounded model checking (BMC) techniques, to detect memory safety and concurrency errors. We also investigate fragility aspects related to the UAV communication link. All UAV components (e.g., position, velocity, and attitude control) heavily depend on the communication link. Our preliminary results show that fuzzing and BMC techniques can detect various software vulnerabilities, which are of particular interest to ensure security in UAVs. We were able to perform successful cyber-attacks via penetration testing against the UAV both connection and software system. As a result, we demonstrate real cyber-threats with the possibility of exploiting further security vulnerabilities in real-world UAV software in the foreseeable future.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs), also sometimes referred to as drones, are aircrafts without human pilots on board; they are typically controlled remotely and autonomously and have been applied to different domains (e.g., industrial, military, and education). In 2018, PWC estimated the impact of UAVs on the UK economy, highlighting that they are becoming essential devices in various aspects of life and work in the UK. The application of UAVs to different domains is leading to a GBP 42bn increase in the UK's gross domestic product and 628,000 jobs in its economy [1].

With this ever-growing interest also comes an increasing danger of cyber-attacks, which can pose high safety risks to large airplanes and ground installations, as witnessed at the Gatwick airport in the UK in late 2018, when unknown UAVs flying close to the runways caused disruption and cancellation of hundreds of flights due to safety concerns [2]. Recent studies conducted by the Civil Aviation Authority show that a 2kg UAV can cause critical damage to a passenger jet windscreen [3]. Therefore, it remains an open question whether the Confidentiality, Integrity, and Availability (CIA) triad principles, which is a model designed to guide policies for information security [4], will be maintained during UAVs' software development life-cycle.

UAVs typically demand high-quality software to meet their target system's requirements. Any failures in embedded (critical) software, such as those embedded in avionics, might lead to catastrophic consequences in the real world. As a result, software testing and verification techniques are essential ingredients for developing systems with high dependability and reliability requirements, needed to guarantee both user requirements and system behavior.

Bounded Model Checking (BMC) was introduced nearly two decades ago as a verification technique to refute safety properties in hardware [5]. However, BMC has only relatively recently been made practical, as a result of significant advances in Boolean Satisfiability (SAT) and Satisfiability Modulo Theories (SMT) [5]. Nonetheless, the impact of this technique is still limited in practice, due to the current size (e.g., number of lines of source code) and complexity (e.g., loops and recursions) of software systems. For instance, when a BMC-based verifier symbolically executes a program,

it encodes all its possible execution paths into one single SMT formula, which results in a large number of constraints that need to be checked. Although BMC techniques are effective in refuting properties, they still suffer from the state-space explosion problem [6].

Fuzzing is a successful testing technique that can create a substantial amount of random data to discover security vulnerabilities in real-world software [7]. However, subtle bugs in UAVs might still go unnoticed due to the large state-space exploration, as recently reported by Chaves et al. [8]. Additionally, according to Alhawi et al. [9], fuzzing could take a significant amount of time and effort to be completed during the testing phase of the software development life-cycle in addition to its code coverage issues. Apart from these limitations, fuzzing and BMC can enable a wide range of verification techniques. Some examples include automatic detection of bugs and security vulnerabilities, recovery of corrupt documents, patch generation, and automatic debugging. These techniques have been industrially adopted by large companies, including but not limited to Amazon Web Service (CBMC [10]), Microsoft (SAGE [11]), IBM (Apollo [12]), and NASA (Symbolic PathFinder [13]). For example, the SAGE fuzzer has already discovered more than 30 new bugs in large shipped Windows applications [11]. Nonetheless, an open research question consists of whether these techniques can be useful in terms of correctness and performance to verify UAV applications.

Our research investigates both fuzzing and BMC techniques to detect security vulnerabilities in real-world UAV software automatically. Thus, our main research goal is to allow the development of software systems that are immune to cyber-attacks and thus ultimately improve software reliability. According to the current cyber-attacks profile concerning advanced UAVs, it becomes clear that the current civilian UAVs in the market are insecure even from simple cyber-attacks. To show this point of view, we highlight in our study real cyber-threats of UAVs by performing successful cyber-attacks against different UAV models. These cyber-attacks led to gaining full unauthorized control or causing the UAVs to crash. We show that pre-knowledge of the receptiveness of the UAV system components is all attackers need to know during their reconnaissance phase before exploiting UAV weaknesses.

A. Contributions

Our main contribution is to propose a novel approach for detecting and exploiting security vulnerabilities in UAVs. We leverage the benefit of using both fuzzing and BMC techniques to detect security vulnerabilities hidden deep in the software state-space. In particular, we make three significant contributions:

- Provide a novel verification approach that combines fuzzing and BMC techniques to detect software vulnerabilities in UAV software.
- Identify different security vulnerabilities that UAVs can be susceptible to. We perform real cyber-attacks against different UAV models to highlight their cyber-threats.
- Evaluate a preliminary verification approach called “UAV fuzzer” to be compatible with the type of UAV software used in industry to exploit their vulnerabilities.

Although our current work represents an ongoing research, preliminary results show that fuzzing and BMC techniques can detect various software vulnerabilities, which are of particular interest to UAV security. We are also able to perform successful cyber-attacks via penetration testing against the UAV both connection and software system. As a result, we demonstrate real cyber-threats with the possibility of exploiting further security vulnerabilities in real-world UAV software in the foreseeable future.

B. Organisation

The rest of the paper is organised as follows. Section II describes the UAVs structure and the recent cyber attacks, in addition to other approaches used to verify security in UAVs. Section III introduces UAV software, UAV communication layer and the methodology to verify it using Fuzzing and Bounded Model Checking techniques. Section IV then describes the benchmarks used and presents the results to determine the effectiveness of our approach. Finally, Section V presents our conclusions.



Figure 1. UAV types: multi-rotor (a), fixed wing (b), and single-rotor (c).

II. BACKGROUND

A. Generic Model of UAV Systems

Reg Austin [14] defines UAVs as a system comprising a number of subsystems, including the aircraft (often referred to as a UAV or unmanned air vehicle), its payloads, the Ground Control Station (GCS) (and, often, other remote stations), aircraft launch and recovery subsystems, where applicable, support subsystems, communication subsystems, and transport subsystems. UAVs have different shapes and models to meet the various tasks assigned to them, such as fixed-wing, single rotor, and multi-rotor, as illustrated in Fig. 1. However, their functional structure has a fixed standard, as shown in Fig. 2. Therefore, finding a security vulnerability in one model might lead to exploiting the same vulnerability in a wide range of different systems [15], [16].

B. Cyber-Threats

A cyber-threat in UAVs represents a malicious action by an attacker with the goal of damaging the surrounding environment, or causing financial losses, where the UAV is typically deployed [17]. In particular, with some of these UAVs available to the general public, ensuring their secure operation still remains an open research question, especially when considering the sensitivity of previous cyber-attacks described in the literature [18], [19]. One notable example is the control of deadly weapons as with the US military RQ-170 Sentinel stealth aircraft; it was intercepted and brought down by the Iranian forces in late 2011 during one of the US military operations over the Iranian territory [18]. In 2018, Israel released footage for one of its helicopters shooting down an Iranian replica model of the US hijacked drone [19]. Further interest in UAV cyber-security has been raised following this attack. For example, Nils Rodday [20], a cyber-security analyst, was able to hack UAVs utilized by the police using a man-in-the-middle attack by injecting control commands to interact with the UAV. As a result of previous attacks, UAVs can be a dangerous weapon in the wrong hands. Obviously, cyber-attack threats exceeded the cyber-space barrier as observed by Tarabay, Lee and Frew [18], [19]. Therefore, enhancing the security and resilience of UAV software has become a vital homeland security mission, mainly due to the possible damage of cyber-attacks from the deployed UAVs.

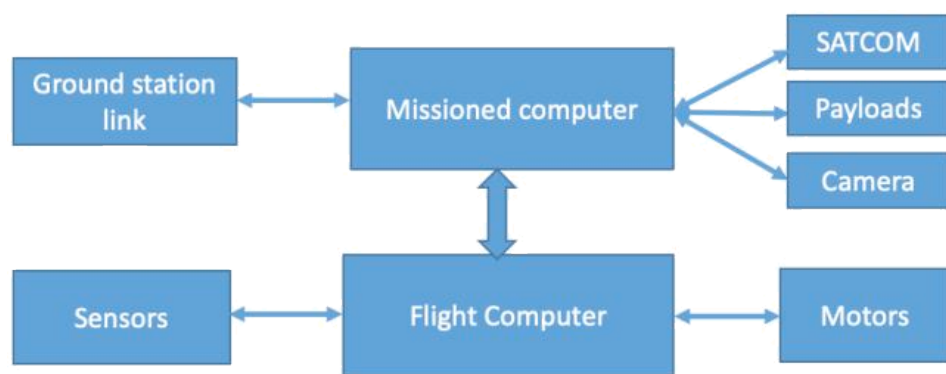


Figure 2. Functional structure of UAVs.

C. Verification of Security in UAVs

The UAV components are typically connected together to enable secure and fast communication; if one component fails, the entire system can be susceptible to malicious attacks [14]. In this respect, various approaches have been taken to automatically verify the correctness of UAV software. In particular, following the RQ-170 UAV accident in 2011, where Iran claimed hacking the sophisticated U.S. UAV [21], a group of researchers from the University of Texas proposed an unusual exercise to the U.S. Department of Homeland Security; specifically, simulated GPS signals were transmitted over the air from 620 m, where the spoofer induced the capture GPS receiver to produce position and velocity solutions, which falsely indicated the UAV position [22]. A similar study conducted in early 2018 performed a successful side-channel attack to leverage physical stimuli [23]. The authors were able to detect in real-time whether the UAV's camera is directed towards a target or not, by analyzing the encrypted communication channel, which was transmitted from a real UAV. These prior studies were able to highlight GPS weaknesses, but they did not cover the UAV security issues w.r.t. all involved software elements, mainly when zero-day vulnerabilities are associated with the respective UAV outputs, i.e., a real UAV software bug that is unknown to the vendor responsible for patching or otherwise fixing the bug. Other related studies focus on automated testing [24] and model-checking the behavior of

UAV systems [8]. For example, a verification tool named as Digital System Verifier (DSVerifier) [8] formally checks digital-system implementation issues, in order to investigate problems that emerge in digital control software designed for UAV attitude systems (i.e., software errors caused by finite word-length effects). Similar work also focuses on low-level implementation aspects, where Sirigineedi et al. [25] applied a formal modeling language called SMV to multiple-UAV missions by means of Kripke structures and formal verification of some of the mission properties typically expressed in Computational Tree Logic. In this particular study, a deadlock has been found and the trace generated by SMV has been successfully simulated. Note that these prior studies concentrate mainly on the low-level implementation aspects of how UAVs execute pilot commands. By contrast, we focus our approach on the high-level application of UAVs software, which is typically hosted by the firmware embedded in UAVs.

Despite the previously discussed limitations, BMC and Fuzzing techniques have been successfully used to verify the correctness of digital circuits, security, and communication protocols [25], [26]. However, given the current knowledge in ensuring security of UAVs, the combination of fuzzing and BMC techniques have not been used before for detecting security vulnerabilities in UAV software (e.g., buffer overflow, dereferencing of null pointers, and pointers pointing to unallocated memory regions). UAV software is used for mapping, aerial analysis and to get optimized images. In this study, we propose to use both techniques to detect security vulnerabilities in real-world UAV software.

III. FINDING SOFTWARE VULNERABILITIES IN UAVS USING SOFTWARE VERIFICATION

A. Software In-The-Loop

UAV software has a crucial role to operate, manage, and provide programmatic access to the connected UAV system. In particular, before a given UAV starts its mission, the missioned computer, as illustrated in Fig. 2, exports data required for this mission from a computer running the flight planning software. Then, the flight planning software allows the operator to set the required flight zone (way-point mission engine), where the UAV will follow this route throughout its mission instead of using a traditional remote controller directly [24].

Dronekit is an open-source software project that allows one to command a UAV using Python programming language [27]; it enables the pilot to manage and direct control over the UAV movement and operation, as illustrated in Fig. 3, where one can connect to the UAV via a User Datagram Protocol (UDP) endpoint (line 3) with the goal of gaining control of the UAV by means of the “vehicle” object. In particular, UDP allows establishing a low-latency and loss-tolerating connection between the pilot and the UAV. This control process relies on the planning software inside the UAV’s system, which in some cases might be permanently connected to the pilot controlling system (e.g., Remote Controller or GCS) due to live feedback or for real-time streaming.

```
1 from dronekit import connect
2 # Connect to UDP endpoint.
3 vehicle = connect('127.0.0.1:14550', wait_ready=True)
4 # Use returned Vehicle object to query device state:
5 print("Mode: %s" % vehicle.mode.name)
```

Figure 3. Python script to connect to a vehicle (real or simulated).

Our main research goal is to investigate in depth open-source UAV code (e.g., DJI Tello and Parrot Bebop) to search for potential security vulnerabilities. For example, Fig. 4 shows a simple Python code to read and view various data status of Tello UAV. In particular, this Python code imports and defines the required libraries (lines from 1 to 3) and then connects the GCS to the UAV by using the predefined port and IP address in lines 11 and 12. As we can see from lines 15 to 25, the UAV will acknowledge the pilot commands and print the Tello current status. If an attacker is able to scan and locate the IP address that this particular UAV has used, then he/she would be able to easily intercept the data transmitted, inject a malicious code or take the drone out of service using a denial of service attack, which can lead the UAV to a crash, thus making it inaccessible. In order to detect potential security vulnerabilities in UAV software, we provide here an initial insight of how to combine BMC and fuzzing techniques with the goal of exploring the system state-space to ensure safe and secure operations of UAVs.

Illustrative Example Using UAV swarm: Throughout this paper, we use an illustrative example from UAV swarm, which consists of multiple UAVs autonomously making decisions based on shared information; the safe and secure operation of multiple UAVs is particularly relevant since they have the potential to revolutionize the dynamics of conflict. In early 2019, we participated in competitive exercises with five different UK universities; the main goal of this event consisted of teams from across the UK competing against each other in a game of offense (red team) and defense (blue team) using swarms of UAVs, as illustrated in Fig. 5. As a result, this competition allowed us to highlight aspects of how to protect urban spaces from UAV swarms, which is a serious concern of modern society. This competition was sponsored by a British multinational defense, security, and aerospace company (BAE).

```

1 import socket \label{python:import11}
2 from time import sleep
3 import curses
4 INTERVAL = 0.2
5 ...
6 local_ip = ''
7 local_port = 8890
8 socket=socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
9 # socket for sending cmd
10 socket.bind((local_ip, local_port))
11 tello_ip = '192.168.10.1'
12 tello_port = 8889
13 tello_adderss = (tello_ip, tello_port)
14 socket.sendto('command'.encode('utf-8'), tello_adderss)
15 try:
16     index = 0
17     while True:
18         index += 1
19         response, ip = socket.recvfrom(1024)
20         if response == 'ok':
21             continue
22         out = response.replace(';', '\n')
23         out = 'Tello_State:\n' + out
24         report(out)
25         sleep(INTERVAL)
26 ...

```

Figure 4. Python code fragment to read and view various data status of Tello UAV.

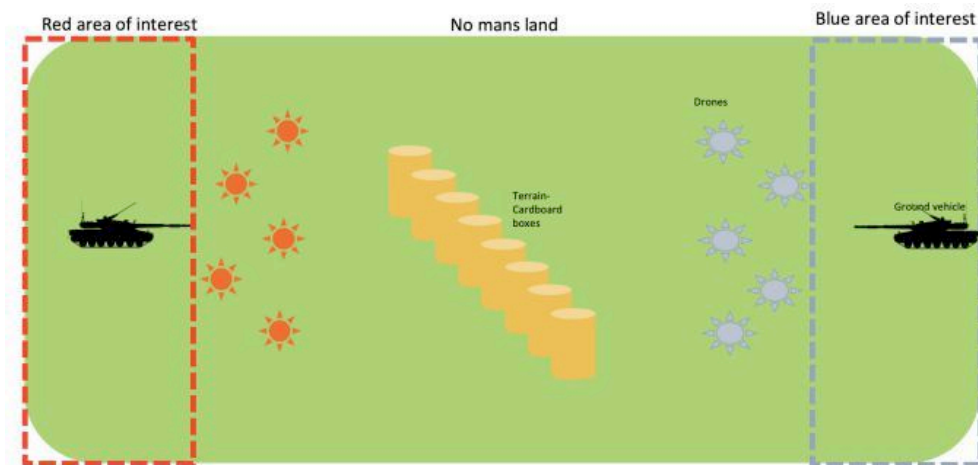


Figure 5. UAV Swarm Competition.

Solutions developed by industry, such as the “jamming guns” and single “UAV catchers”, fall short of what would be required to defend against a large automated UAV swarm attack. For this particular illustrative example, using software verification and the UAV connection weakness, we were able to perform a successful cyber-attack against UAV models by scanning the radio frequencies and targeting the unwanted UAVs with just a Raspberry-pi with a Linux OS installed and 2:4 GHz antennas, as reported in our experimental evaluation.

B. Verifying UAV Software using Fuzzing and BMC

We describe our novel verification approach, called “UAV Fuzzer”, to check for security vulnerabilities in UAVs. In particular, we check for user-specified assertions, buffer overflow, memory safety, division by zero, and arithmetic under- and overflow. Our verification approach consists of running a fuzzer engine using pre-collected test cases $TC = \{Tc_1; Tc_2; \dots; Tc_n\}$, where n represents the total number of pre-collected test cases, with the goal of initially exploring the state-space of the UAV software operation. Note that a test case tc_i used by our approach is similar to valid data, but it must contain a problem on it, also called “anomalies”. For example, to fuzz UAV software, a test case should be a connection between the UAV and GCS, so the mutated version generated of such a similar connection is called a test case tc .

In our “UAV Fuzzer”, we keep track of each computation path of the program $\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$, which represents the program unfolding for a bound k and a security property φ initially explored by our fuzzer. If our fuzzer engine gets stuck due to the mutations generated not being suited enough to the new state transition, our BMC tool runs against the target software to symbolically explore its uncovered state-space with the goal of checking the unexplored execution paths in Π of the UAV software. The idea behind BMC is to check the negation of a given property φ at a given depth k , i.e., given a transition system M , a property φ , and a limit of iterations k , BMC unfolds a given system k times and converts it into a Verification Condition (VC) ψ , such that ψ is *satisfiable* if and only if φ has a counterexample (*cex*) of depth less than or equal to k .

We formally describe our verification algorithms “UAV Fuzzer” by assuming that a given program P under verification is a state transition system M . In M , a state $s \in S$ consists of the value of the program counter and the values of all program variables. A predicate $init_P(s)$ denotes that s is an initial state, $tr_P(s_i, s_j) \in T$ is a transition relation from s_i to s_j , $\varphi(s)$ is the formula encoding for states satisfying a safety property, and $\psi(s)$ is the formula encoding for states satisfying a completeness threshold [28], which is equal to the maximum number of loop iterations occurring in P . For convenience, we define an error state ε , reachable if a property violation exists in the program P . A counterexample cex^k is a sequence of states of length k from an initial state s_i to ε . The main steps for our proposed verification algorithm are described in Algorithm 1.

As an illustrative example, consider the code fragment shown in Fig. 4. First, our UAV fuzzer starts by defining new data based on the input expected by our targeted model (Tello UAV). As an example, Fig. 6 shows a valid test case generated by our UAV fuzzer, which is based on the module specification for the Tello UAV; this test case expects the IP address and specific port before launching the drone to start flying. Second, our UAV fuzzer engine starts exploring and running the UAV software (cf. Fig. 4) with the generated test-cases (cf. Fig. 6) and then it records each execution path π_i that has been explored. Third, when the UAV fuzzer engine reaches a complex condition and struggles to find its next path (line 5 of Fig. 6), UAV fuzzer will attempt to reconstruct the following path using BMC, which stores the current fuzzing transactions and restores the next path symbolically. Lastly, BMC will check for any further exception that occurs as a result of its execution. Additionally, UAV fuzzer can report the code coverage achieved during the

testing process, and thus provide a better understanding of code coverage status.

Algorithm 1 UAV Fuzzer

- 1: Define pre-collected test cases $TC = \{tc_1, tc_2, \dots, tc_n\}$ to be employed by the fuzzing engine.
 - 2: Fuzzer engine begins to explore each execution path π_i starting from an initial state s and produces malformed inputs $I = \{\iota_1, \iota_2, \dots, \iota_n\}$ to test for potential security vulnerabilities.
 - 3: Store each π_i that has been verified and repeat step 2 until the fuzzer engine either reaches a crashing point or it cannot explore the next π_i in Π due to complex guard checks.
 - 4: Run BMC to verify the remaining execution paths in Π that have not been previously explored by our fuzzer engine in steps 2 and 3.
 - 5: Repeat step 4 until BMC falsifies or verifies ϕ or it exhausts time and memory limits.
 - 6: Once BMC completely verifies the UAV code in step 5, it returns “false” if a property violation is found together with cex^k , “true” if it is able to prove correctness, or “unknown”.
-

```

1  while True:
2      index %= 1
3      # + replaced with %
4      response, ip = socket.recvfrom(1024)
5      if response == 'ok':
6          continue
    
```

Figure 6. Test case from the Tello UAV embedded software.

C. UAV Communication Channel

A UAV has a radio to enable and facilitate remote communication between the GCS and the UAV. In addition, it consists of different electronic components, which interact autonomously with a goldmine of data transmitted over the air during its flight missions; this makes the communication channel in UAVs an ideal target for a remote cyber-threat. Therefore, ensuring secure (bug-free) software, together with a secure communication channel, emerges as a priority in successful deployment of any UAV system.

A successful false-data injection attack was demonstrated by Strohmeier et al. [9], which had devastating effects on the UAV system. The authors were able to successfully inject valid-looking messages, which are well-formed with reasonable data into the Automatic Dependent Surveillance-Broadcast (ADS-B) protocol. Note that this protocol is currently the only means of air traffic surveillance today, where Europe and US must comply with the mandatory use of this insecure protocol by the end of 2020 [29].

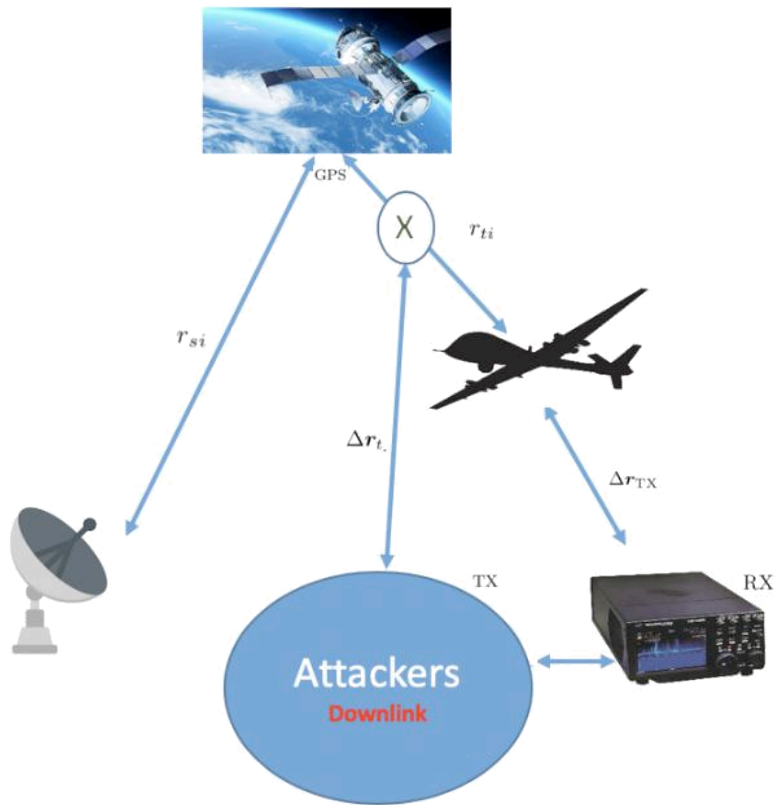


Figure 7. GPS-satellite-signal is overlaid by a spoofed GPS-signal.

To investigate this layer further, we used a Software-Defined Radio (SDR) system to receive, transmit, and analyze the UAV operational connection system (e.g., Ku-Band and WiFi). We have also investigated the information exchanged between UAV sensors and the surrounding environment for any potential security vulnerabilities (e.g., GPS Spoofing), as illustrated in Fig. 7. The signal that comes from the satellite is weak; hence, if an attacker uses a local transmitter under the same frequency, this signal would be stronger than the original satellite signal. As a result, the spoofed GPS-signal will override the current satellite-signal, thereby leading to the ability to spoof a fake position for the UAV targeted. In this particular case, the UAV would then be hijacked and put in hold, waiting for the attacker's next command. Therefore, verifying the UAV software to build practical software systems with strong safety and security guarantees is highly needed in practice.

Our GPS spoofing attack is described in Algorithm 2. Here, spoofer refers to a Full-Duplex device that is used to attack a particular UAV system to crash or take control of the UAVs. In our experimental evaluation with the UAV models DJI Tello and Parrot Bebop 2, we were able to perform a successful attack under 2.4 GHz to the target system. First, we were able to detect the drone frequency by one of the antennae, which was configured to monitor the active 2.4 GHz connection referred to as RX. The targeted drones are allocated based on identifying the drone default MAC addresses owned by the Parrot company and their unique SSID. Second, the other antenna was used to transmit the attacker new data referred to as TX. The distance between the attacker equipment and the target (r_{si} and r_{ti}) is vital for this attack, since the antenna/system strength and the delay caused are all taken into consideration during the attack.

Algorithm 2 GPS Spoofing Attack

- 1: The spoofer device should be located with the nominated antennas (i.e., 2.4 GHz).
- 2: The antenna configured on monitoring mode (RX) to detect the authentic signal from the available GPS satellites.
- 3: The vectors $\Delta \mathbf{r}_{TX}$ and $\Delta \mathbf{r}_t$ are for the antenna (TX) coordinates which distributed in a 3-dimensional array.
- 4: r_{si} and r_{ti} are the respective distances from the GPS satellite.

IV. PRELIMINARY EXPERIMENTAL EVALUATION

We have performed a preliminary evaluation of our proposed verification approach to detect security vulnerabilities in UAVs. Our proposed method was implemented in a tool called DepthK [30], using BMC technique and invariant generators such as PIPS [31] and PAGAI [32]. PIPS is an inter-procedural source-to-source compiler framework for C and Fortran programs, PAGAI is a tool for automatic static analysis that is able to generate inductive invariants, both rely on a polyhedral abstraction of program behavior for inferring invariants. We have evaluated the DepthK tool over a set of standard C benchmarks, which share common features of UAV code (e.g., concurrency and arithmetic operations). We have also evaluated our fuzzer engine to test PDF software with the goal of checking its efficiency and efficacy to identify bugs. Lastly, we present our results in the swarm competition promoted by BAE systems.

A. Description of Benchmarks

The International Software Verification Competition (SV-COMP) [33], where DepthK participated, was run on a Linux Ubuntu 18:04 OS, 15 GB of RAM memory, a run-time limit of 15 minutes for each verification task and eight processing units of an i7 4790 CPU. The SV-COMP's benchmarks used in this experimental evaluation include: Reach-Safety, which contains benchmarks for checking reachability of an error location; MemSafety, which presents benchmarks for checking memory safety; ConcurrencySafety, which provides benchmarks for checking concurrency problems; Overflows, which is composed of benchmarks for checking whether variables of signed-integers type overflow; Termination, which contains benchmarks for which termination should be decided; SoftwareSystems, which provides benchmarks from real software systems.

Our fuzzing experiments were run on a MacBook Pro laptop with 2.9 GHz Intel Core i7 processor and 16 GB of memory. We ran our fuzzing engine for at most of 12 hours for each single binary file. We analyzed and replayed the testing result after a crash was reported or after the fuzzer hit the time limit. To analyze the radio frequencies, we configured/compiled the required software for this purpose (e.g. bladerf, GQRX, OsmoSDR, and GNU Radio tool) using bladerf x40 device, ALFA high gain USB Wireless adapter and 2.4 GHz antennas. Additionally, we used the open-source UAVs code DJI Tello and Parrot Bebop.

B. Objectives

The impact of our study is a novel insight on the UAV security potential risks. In summary, our evaluation has the following three experimental questions to answer:

- EQ1 (Localization) Can DepthK help us understand the security vulnerabilities that have been detected?
- EQ2 (Detection) Can generational or mutational fuzzers be further developed to detect vulnerabilities in real-world software?
- EQ3 (Cyber-attacks) Are we able to perform successful cyber-attacks in commercial UAVs?

C. Results

1. **SV-COMP:** Concurrency bugs in UAVs are one of the most difficult vulnerabilities to verify [25]. Our software verifier DepthK [30] has been used to verify and falsify safety properties in C programs, using BMC and k-induction proof rule techniques. In late 2018, we participated with the DepthK tool in SV-COMP 2019 against other software verifiers. Our verifier showed promising results over thousands of verification tasks, which are of particular interest to UAVs security (e.g., Concurrency Safety and Overflows categories), which answers EQ1. Concurrency Safety category, which consists of 1082 benchmarks of concurrency problems, is one of the many categories verifiers run over; DepthK was able to accurately detect 966 problems from this category. For the Overflows category, which consists of 359 benchmarks for different signed-integers overflow bugs, DepthK was able to detect 167 problems. These results are summarized in Table I. A task counts as correct true if it does not contain any reachable error location or assertion violation, and the tool reports “safe”; however, if the tool reports “unsafe”, it counts as incorrect true. Similarly, a task counts as correct false if it does contain a reachable violation, and the tool reports “unsafe”, together with a confirmed witness (path to failure); otherwise, it counts as incorrect false accordingly. Dirk Beyer [33] shows DepthK’s results when compared with other verifiers in SV-COMP 2019.

Table I DEPT HK RESULTS IN SV-COMP 2019.

Category list	Correct True	Correct False	Incorrect Results	Unknown
Concurrency Safety	194	772	20	96
Overflows	17	150	0	192

Table II FUZZING APPROACHES COMPARISON.

Fuzzing Approaches	Target	Time	Faults
Generational Fuzzer	Sumatra PDF	45 hours	70
Mutational Fuzzer	Sumatra PDF	15 hours	23

2. **Fuzzing Approach:** According to a prior study [9], the generalizing fuzzing approach leads to a better result in discovering and recording software vulnerabilities compared with the mutational fuzzing approach if the test cases used in the fuzzing experiment are taken into account, which answers EQ2. Our experimental results applied to a PDF software called Sumatra PDF, which was chosen for evaluation purposes, are shown in Table II. Here, the generational fuzzer was able to detect 70 faults in 45 hours in the Sumatra PDF, while the mutational fuzzer was able to detect 23 in 15 hours.
3. **UAV Swarm Competition:** As part of our participation at the UAV swarm competition sponsored by BAE, penetration testing was performed against both the UAV connection and software system, in which we were able to perform successful cyber-attacks, which answers EQ3. These attacks led us to deliberately crash UAVs or to take control of different non-encrypted UAV systems (e.g., Tello and Parrot Bebop 2). This was achieved by sending connection requests to shut down a UAV CPU, thereby sending packets of data that exceed the capacity allocated by the buffer of the UAV's flight application and by sending a fake information packet to the device's controller. These results are summarized in Table III, where we describe the employed UAV models and tools and whether we were able to obtain full control or crash. Note that due to the limitations of the competition, DepthK tool was not employed during the BAE competition; however, exploiting potential UAV software vulnerabilities is still a continuous research, where we intend to further exploit DepthK.

D. Threats to Validity

Benchmark selection: We report the evaluation of our approach over a set of real-world benchmarks, where the UAVs share the same component structure. Nevertheless, this set is limited within our research scope and the experiment results may not generalize to other models because other UAV models have a proprietary source-code. Additionally, we have not evaluated our verification approach using real UAV code written in Python, which is our main goal for future research.

Table III RESULTS OF THE UAV SWARM COMPETITION.

Vulnerability Type	UAV Model	Tool	Result
Spoofing	DJI Tello	Wi-Fi transmitter	Full Control
Denial of service			Full Control
Spoofing	Parrot bebop 2	Wi-Fi transmitter	Full Control
Denial of service			Crash

Radio Spectrum: The frequencies we report on our evaluation were between 2:4 GHz and 5:8 GHz, as the two most common ranges for civilian UAVs; however, the radio regulations in the UK are complicated (e.g., we are required to be either licensed or exempted from licensing for any transmission over the air).

IV. CONCLUSIONS AND FUTURE WORK

Our ultimate goal is to develop a UAV fuzzer to be introduced as mainstream into the current UAV programming system, in order to build practical software systems robust to cyber-attacks. We have reported here an initial insight of our verification approach and some preliminary results using similar software typically used by UAVs. In order to achieve our ultimate goal, we have various tasks planned as follows:

- **Vulnerability Assessment:** Identify and implement simple cyber-attacks from a single point of attack against different UAV models. We will continue investigating Python vulnerabilities at the high-level system (e.g., UAV applications) and whether UAV software is exploitable to those security vulnerabilities.
- **Python Fuzzer:** We will develop an automated Python fuzzer by analyzing how to convert the UAV command packets into fuzzing ones, in order to produce test cases that are amenable to our proposed fuzzer.
- **GPS Analysis:** We identified based on numerical analysis on GPS, the cyber-attack UAVs might be vulnerable to. This investigation will continue to develop and simulate a GPS attack applied to a real UAV system.
- **Implementation:** Apply our proposed verification approach to test real-world software vulnerabilities, which can be implemented during the software development life-cycle to design a cyber-secure architecture.
- **Evaluation and Application:** Evaluate our proposed approach using real-world UAV implementation software. We will also compare our approach in different stages to check its effectiveness and efficiency.

REFERENCES:

1. PwC, “Drones could add £42bn to UK GDP by 2030 - PwC research,” 2018. [Online]. Available: <https://www.pwc.co.uk/press-room/press-releases/pwc-uk-drones-report.html>
2. [2] PwC., “Gatwick airport drones disruption wasn’t all for nothing, UK police insist,” 2018. [On-line]. Available: <https://edition.cnn.com/2018/12/24/uk/gatwick-airport-drones-investigation-gbr-intl/index.html>
3. A. House, Drone Safety Risk: An assessment. Civil Aviation Authority, 2018.
4. M. Farooq, M. Waseem, A. Khairi, and S. Mazhar, “Article: A critical analysis on the security concerns of internet of things (iot),” IJCA, vol. 111, no. 7, pp. 1–6, February 2015.
5. A. Biere, Handbook Of Satisfiability. IOS Press, 2009, vol. 185, ch. 26.
6. M. R. Gadelha, F. R. Monteiro, J. Morse, L. C. Cordeiro, B. Fischer, and D. A. Nicole, “ESBMC 5.0: an industrial-strength C model checker,” in ASE, 2018, pp. 888–891. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3238147.3240481>
7. B. P. Miller, G. Cooksey, and F. Moore, “An empirical study of the robustness of macos applications using random testing,” in RT. ACM, 2006, pp. 46–54.
8. L. Chaves, I. Bessa, H. Ismail, A. Frutuoso, L. Cordeiro, and E. de Lima Filho, “DSVerifier-Aided Verification Applied to Attitude Control Software in Unmanned Aerial Vehicles,” IEEE Transactions on Reliability, vol. 67, 2018.
9. O. Alhawi, A. Akinbi, and A. Dehghantanha, “Evaluation and Application of Two Fuzzing Approaches for Security Testing of IoT Applications,” in Handbook of Big Data and IoT Security, 2019, pp. 301–327. [Online]. Available: http://link.springer.com/10.1007/978-3-030-10543-3_f_g13
10. B. Cook, K. Khazem, D. Kroening, S. Tasiran, M. Tautschnig, and M. R. Tuttle, “Model checking boot code from AWS data centers,” in Computer Aided Verification, vol. 10982. Springer, 2018, pp. 467–486.
11. P. Godefroid, M. Y. Levin, and D. Molnar, “Automated Whitebox Fuzz Testing,” Queue - Networks, 2012. [Online]. Available: https://www.eecs.northwestern.edu/f_grobby/courses/395-495-2017-winter/ndss2008.pdf
12. S. Artzi, A. Kiezun, J. Dolby, F. Tip, D. Dig, A. Paradkar, and M. D. Ernst, “Finding Bugs in Dynamic Web Applications,” ISSTA, pp. Pages 261–272, 2008. [Online]. Available: <http://www.htmlkit.com>
13. Corina S. Pasareanu, “Using Symbolic (Java) Pathfinder at NASA,” Nasa, 2010. [On-line]. Available: <https://pdfs.semanticscholar.org/8933/ac2dbeb3ccd8cf3e393d8dbb22ccc4452b64.pdf>

REFERENCES:

14. R. Austin, "UNMANNED AIRCRAFT SYSTEMS UAVS DESIGN, DEVELOPMENT AND DEPLOYMENT," A John Wiley and Sons, vol. 54, 2011. [Online]. Available: <https://archive.org/details/UnmannedAircraftSystemsUAS>
15. V. Dey, V. Pudi, A. Chattopadhyay, and Y. Elovici, "Security vulnerabilities of unmanned aerial vehicles and countermeasures: An experimental study," in VLSID. IEEE Computer Society, 2018, pp. 398–403.
16. S. Frei, M. May, U. Fiedler, and B. Plattner, "Large-scale vulnerability analysis," in LSAD '06, 2006, pp. 131–138.
17. A. Y. Javaid, W. Sun, V. K. Devabhaktuni, and M. Alam, "Cyber security threat analysis and modeling of an unmanned aerial vehicle system," in HST, Nov 2012, pp. 585–590.
18. Joanna Frew, "An overview of new armed drone operators The Next Generation," Drone Wars UK, Oxford, Tech. Rep., 2018. [Online]. Available: www.dronewars.net
19. O. L. Jamie Tarabay and I. Lee, "Israel: Iranian drone we shot down was based on captured US drone - CNN," 2018. [Online]. Available: <https://edition.cnn.com/2018/02/12/middleeast/israel-iran-drone-intl/>
20. N. M. Rodday, R. d. O. Schmidt, and A. Pras, "Exploring security vulnerabilities of unmanned aerial vehicles," in NOMS, April 2016, pp. 993–994.
21. A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via gps spoofing," Journal of Field Robotics, vol. 31, no. 4, pp. 617–636, 2014. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21513>
22. M. L. Psiaki, B. W. O'Hanlon, J. A. Bhatti, D. P. Shepard, and T. E. Humphreys, "Gps spoofing detection via dual-receiver correlation of military signals," IEEE Transactions on Aerospace and Electronic Systems, vol. 49, no. 4, pp. 2250–2267, OCTOBER 2013.
23. B. Nassi, R. Ben-Netanel, A. Shamir, and Y. Elovici, "Drones' cryptanalysis - smashing cryptography with a flicker," in 2019 IEEE Symposium on Security and Privacy (SP). Los Alamitos, CA, USA: IEEE Computer Society, may 2019, pp. 832–849. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/SP.2019.00051>
24. M. A. Day, M. R. Clement, J. D. Russo, D. Davis, and T. H. Chung, "Multi-uav software systems and simulation architecture," in 2015 International Conference on Unmanned Aircraft Systems (ICUAS), June 2015, pp. 426–435.
25. G. Sirigineedi, A. Tsourdos, R. Zbikowski, and B. A. White, "Modelling and Verification of Multiple UAV Mission Using SMV," EPTCS, vol. 20, pp. 22–33, 2010. [Online]. Available: <https://dspace.lib.cranfield.ac.uk/handle/1826/3978>



Cyber Attack Vulnerabilities Analysis for UAV

Brandon Wampler

James Goppert

Inseok Hwang



ABOUT THE AUTHOR

BRANDON WAMPLER

B.S. in Aeronautical and Astronautical Engineering from Purdue University



ABOUT THE AUTHOR

INSEOK HWANG

Dr. Hwang is a Professor in the School of Aeronautics and Astronautics at Purdue University. He is a member of Purdue's Signature Area for System-of-Systems research. He earned his Ph.D. degree, specialized in the area of multiple-vehicle control and its application to air traffic control using hybrid systems approach, in the Department of Aeronautics and Astronautics at Stanford University.



ABOUT THE AUTHOR

JAMES GOPPERT

I work with autopilot development, software verification and validation, and simultaneous localization and mapping. I was born in Missouri. I have a B.S/M.S. from Purdue University in Aeronautics and Astronautics. I like to sail in my free time.

As the technological capabilities of automated systems have increased, the use of unmanned aerial vehicles (UAVs) for traditionally exhausting and dangerous manned missions has become more feasible. The United States Army, Air Force, and Navy have released plans for the increased use of UAVs, but have only recently shown interest in the cyber security aspect of UAVs. As a result, current autopilot systems were not built with cyber security considerations taken into account, and are thus vulnerable to cyber attack. Since UAVs rely heavily on their on-board autopilots to function, it is important to develop an autopilot system that is robust enough to thwart possible cyber attacks. In order to develop a cyber-secure autopilot architecture, we have run a study on potential cyber threats and vulnerabilities of the current autopilot systems. This study involved a literature review on general cyber attack methods and on networked systems, which we used to identify the possible threats and vulnerabilities of the current autopilot system. We then studied the identified threats and vulnerabilities in order to analyze the post-attack behavior of the autopilot system through simulation. The uses of UAVs are increasing in many applications other than the traditional military use. We describe several example scenarios involving cyber attacks that demonstrate the vulnerabilities of current autopilot systems.

I. INTRODUCTION

I.A. Background

The use of unmanned aerial vehicles (UAVs) has been limited to military use for the past decade. Some of the missions UAVs have been used for in that context include:

- Surveillance
- Reconnaissance
- Tracking
- Combat
- Support

Research conducted by Frost and Sullivan between 2004 and 2008 shows that the number of UAVs deployed globally on operations has increased from around 1,000 to 5,000 systems. [1] The growth in the use of UAVs will continue as the technology improves, leading to cheaper and more capable unmanned systems. Some of the non-military uses of UAVs are highlighted in Figure 1. As the number of UAVs in use increases, the potential for and interest in cyber attacks on the UAVs also increases. Some of the general unmanned systems problems have already been identified using the assumption that the unmanned system is a remote node of a network, as in [2], but there has not been any research focusing on UAV autopilot vulnerabilities specifically.

The interest in UAV cyber security has been raised greatly after the Predator UAV video stream hijacking incident in 2009, where militants used cheap, off-the-shelf equipment to stream video feeds from a UAV. With greater funding and skills, the possible damage due to a cyber attack is a major concern. The U.S. Army, Air Force, and Department of Defense have all shown clear interest in defending against cyber attacks on deployed UAVs. In depth research into cyber attack threats and vulnerability identification on these systems is therefore required. [3-5]

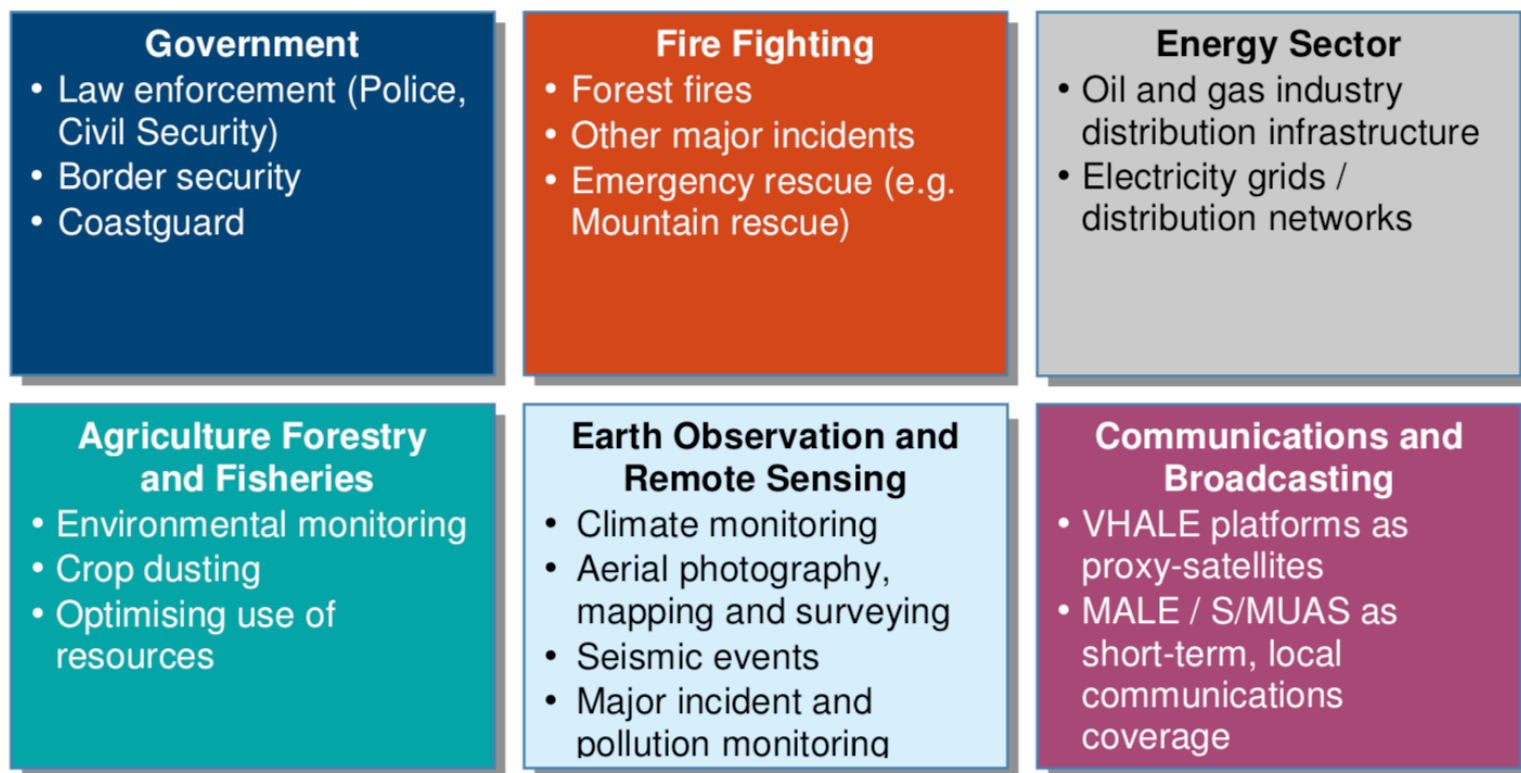


Figure 1: List of future uses for UAVs [1]

I.B. Scope

Our ultimate goal is to develop a cyber secure autopilot architecture for UAVs. For this initial study, we focused on:

1. Identification of the potential threats and vulnerabilities of the current autopilot system.
2. Analysis of post-attack behavior of the UAV.

I.C. Assumptions

For this research we assume that it is possible to corrupt data within the data flow of the autopilot system through methods such as buffer overflow attacks and other cyber attacks. [6] We do not consider the method of attack, but rather we look at the effect that maliciously corrupted data has on the UAV and the damage that results.

II. Study Methodology

II.A. Threats and Vulnerability Identification

First, a study on the current UAV systems, including components such as ground stations and satellites, was carried out in order to verify the data flow in and out of the UAV itself. Then, using our knowledge of the current autopilot system, we hypothesized ways of corrupting data in the autopilot data flow path. Literature reviews were carried out in order to gauge the possibility of the hypothesized attack methods.

II.B. Post-Attack Behavior Analysis

Using a high fidelity aircraft model, we carried out extensive numerical analysis in order to study the post-attack behavior of the UAV to specific cyber attacks. Sensitivity studies were also performed for each of the different cyber attack scenarios in order to determine the most effective attacks. The aircraft model and numerical analysis method will be covered in detail.

III. Statement of the Problem

The general autopilot system structure has not changed since it was introduced for manned aerial vehicles. Because the autopilot system was originally developed for manned systems, cyber security was not a design priority. This makes the current UAV autopilot systems vulnerable to malicious cyber attacks. [7] Our problem is to redesign the autopilot system to be robust enough to thwart potential cyber attacks. In order to do this, we will identify the threats to and vulnerabilities of the current UAV autopilot system in the context of cyber attacks and analyze the post-attack behavior of the UAV.

III. Current and Future UAV Systems

UAVs have been used by the military for over a decade with huge success. Currently, the military is changing their infrastructure to move towards more network-centric warfare, where all of the components of the military are interconnected through sophisticated networks. [8] This will provide fast communication and constant environmental and asset awareness for the entire military. Some UAV systems, such as the Global Hawk, already employ this type of infrastructure, as seen in Figure 2. It is important to notice that all the components of the system are interconnected, and an attack on one component can cause a propagation of failures throughout the whole system. In this project, we focus only on the cyber security of a UAV autopilot, and not on the larger network. The possibility of a UAV cyber attack causing failures in other network components should be noted nonetheless. As the military moves further into network-centric infrastructures and as civilian applications follow, the potential damage of a cyber attack increases.

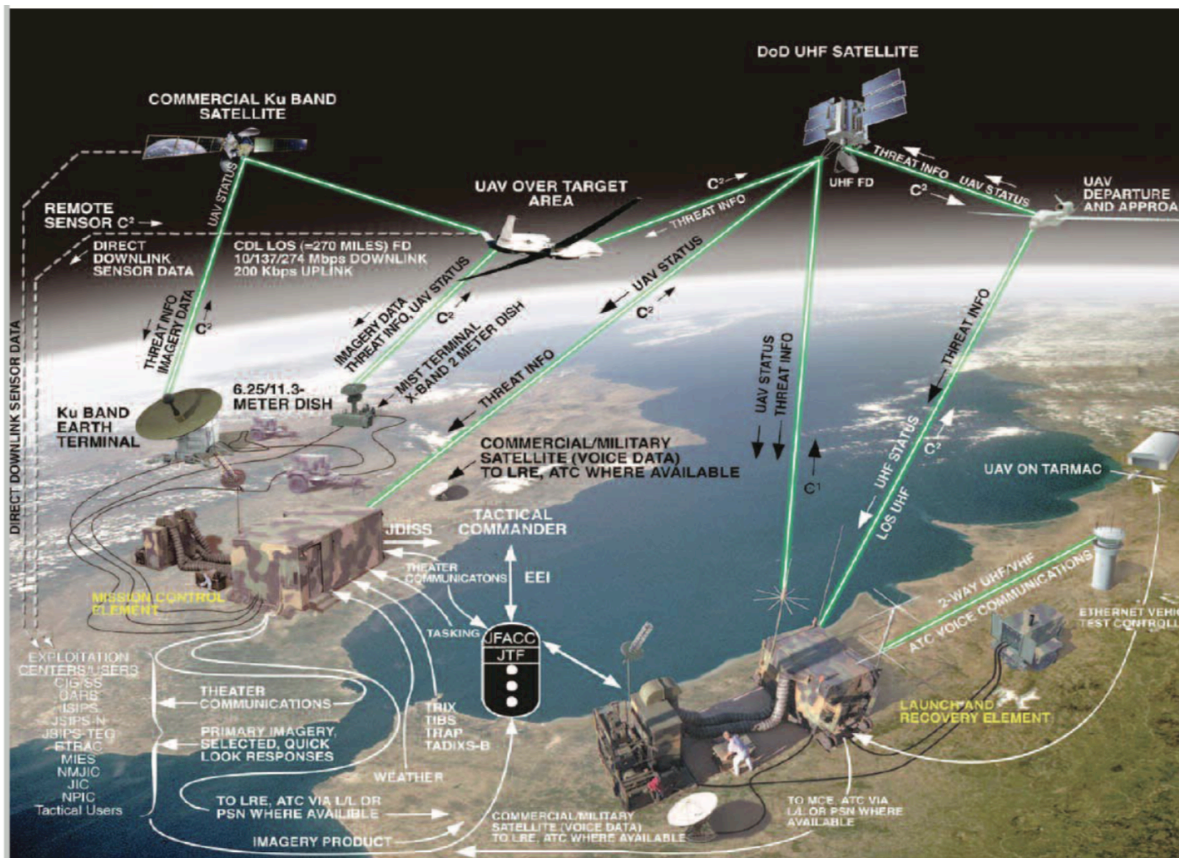


Figure 2: Global Hawk UAV Systems Architecture [9]

V. Current Autopilot Architecture

The common components found in UAV autopilots are:

- **Main Program/Processor:** Responsible for processing sensor data and the implementation of the control of the UAV.
- **Magnetometer:** Used for measuring direction.
- **GPS:** Used to determine the global position.
- **Airspeed/Altimeter:** Used to measure air speed and altitude.
- **UAV Wireless Communication:** Responsible for communicating with the ground station.
- **Power System:** Responsible for providing power to the entire UAV.
- **Inertial Measurement Unit:** Used to measure the movement of the UAV.
- **Boot Loader Reset Switch:** Used to load programs into the main program board.
- **Actuators:** Receives commands from the main processing board and moves the control surfaces.

- **Manual Flight Control:** Overrides the autopilot and gives control of the UAV control surfaces to the ground station.

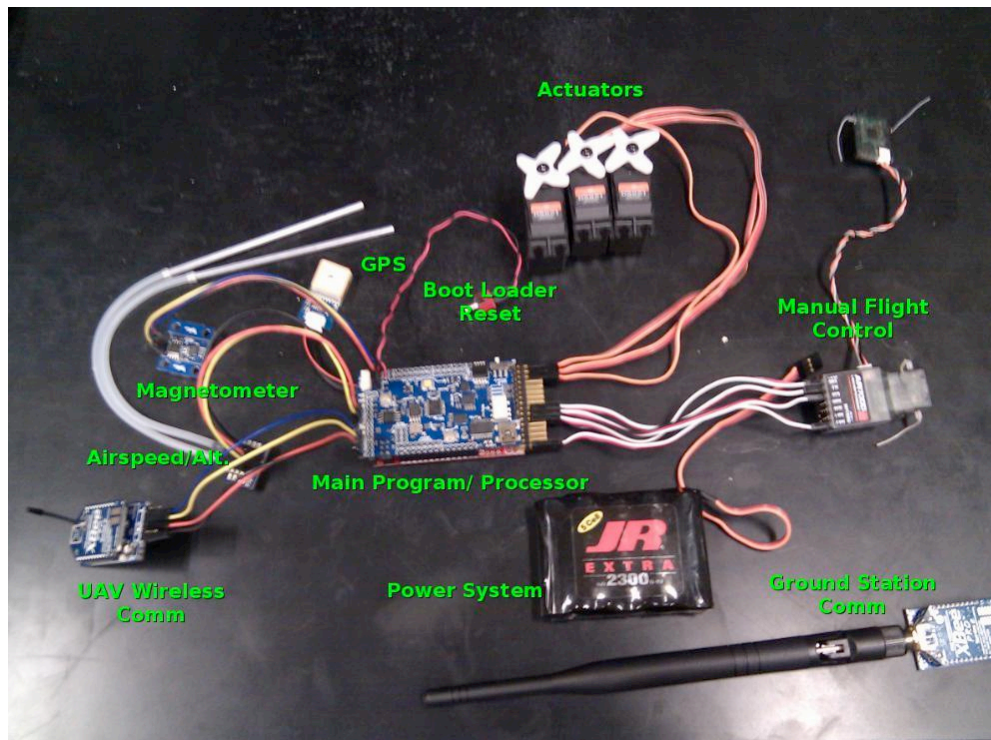


Figure 3: UAV autopilot components used at Purdue Hybrid Systems Lab (ArduPilotMega)

Figure 3 shows the components for the UAV autopilot used in our lab, the Purdue Hybrid Systems Lab. Although the specific equipment shown in Figure 3 is used for a small low cost UAV, it shares the same components found in larger and more expensive UAVs. For example, more expensive UAVs might have more powerful processing units or satellite communication links.

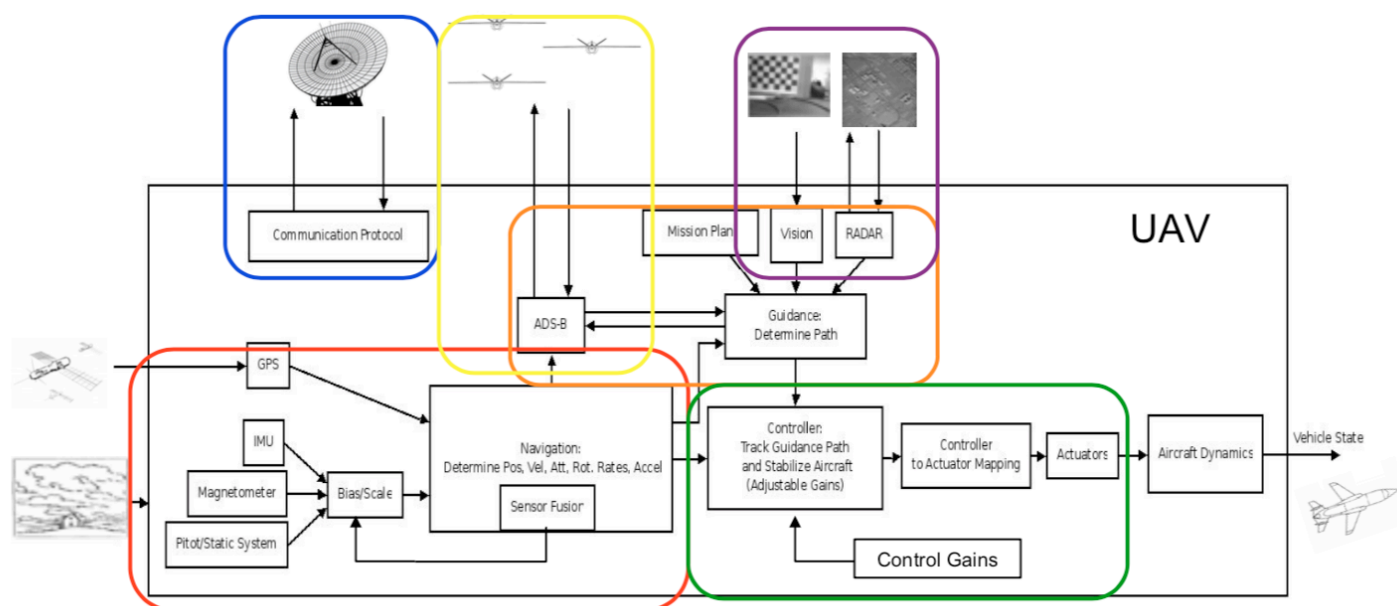


Figure 4: UAV autopilot architecture data flow diagram

Figure 4 shows the UAV autopilot components in a data flow diagram. In this diagram, the components of the autopilot can be divided into three major parts that make up the GNC:

- **Guidance:** Determines a path based on the UAV's state, waypoints, mission objective, avoidance maneuvers, target tracking, etc. (e.g., trajectory generation). Marked in orange in Figure 4.

- **Navigation:** Determines the UAV's state using sensory data. Marked in red in Figure 4.
- **Control:** Keeps the UAV in a safe, stable state in the presence of disturbances and steers the UAV to the target based on information from the Guidance and Navigation. Marked in green in Figure 4.

In Figure 4, the component marked in blue represents the communication in and out of the UAV, the component marked in yellow represents the new ADS-B communication, and the component marked in purple represents the sensors that can aid the guidance of the UAV.

VI. Threat and Vulnerability Identification

VI.A. General Attack Possibilities

We have determined, through studying the data flow in the autopilot, several general cyber attack feasibilities.

These attacks have been categorized into three groups:

- **Hardware Attack:** Attacker has access to the UAV autopilot components directly.
- **Wireless Attack:** Attacker carries out the attacks through one of the wireless communication channels.
- **Sensor Spoofing:** Attacker passes false data through the on-board sensors of the UAV autopilot.

VI.A.1. Hardware Attack

Hardware attacks can occur whenever an attacker has direct access to any of the UAV autopilot components. An attacker can then corrupt the data stored on-board the autopilot or install extra components that can corrupt the data flow. These types of attacks can be carried out during the maintenance and storage of the UAV or during the manufacturing and delivery. An attacker can link directly to the UAV autopilot and damage it or reprogram it if he has the means or replace or add components that will give him control over the UAV and/or the tactical data collected. Hardware attacks can affect the survivability of the UAV, compromise control of the UAV, and compromise the tactical data collected by the UAV.

VI.A.2. Wireless Attack

Wireless attacks can occur if an attacker uses the wireless communication channels to alter data on-board the UAV autopilot. The worst case scenario for this attack is if an attacker is able to break the encryption of the communication channel. Once this occurs, an attacker can gain full control of the UAV if the communication protocol is known. Another possibility is an attack such as a buffer overflow that corrupts some data on-board or initiates some event. The most significant danger of wireless attacks is the fact that an attacker can carry out the attacks from afar while the UAV is being operated.

VI.A.3. Sensor Spoofing

Sensor spoofing attacks are directed towards on-board sensors that depend on the outside environment. Examples of such sensors are the GPS receivers, vision, radar, sonar, lidar, and IR sensors. An attacker can send false data through the GPS channels, or blind any of the vision sensors. The UAV autopilot relies heavily on sensor data for Guidance and Navigation, so corrupted sensor data can be very dangerous.

VI.B. Specific Attack Scenarios

We have further identified specific attack scenarios that show the vulnerabilities of the current autopilot system. Looking at these threats from the cybersecurity perspective, it was natural to further categorize the attacks into the following two types:

- **Control System Security:** Attacks that attempt to prevent the hardware/CPU from behaving as programmed. Some examples of this type of attack include buffer overflow exploits through some input device, forced system resets to load malicious code, and hardware changes or additions to the system.
- **Application Logic Security:** Attacks that use malicious manipulation of the sensors or the environment, providing false data to the control system. In this case, the control system behaves as programmed without fault, but some or all inputs to the system are corrupted. Some examples of this type of attack include sensory data manipulation, vehicle/system component state data manipulation, navigational data manipulation, and command and control (C²) data manipulation. Figure 5 shows the most likely type of vulnerabilities for each component of the UAV autopilot.

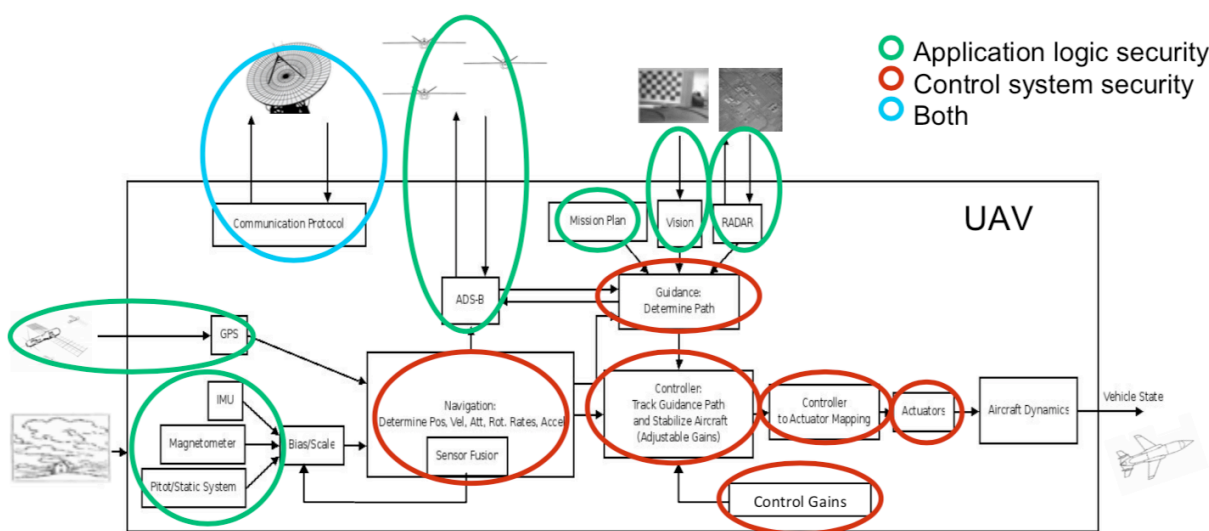


Figure 5: Vulnerabilities of the current UAV autopilot.

VI.B.1. Gain Scheduling

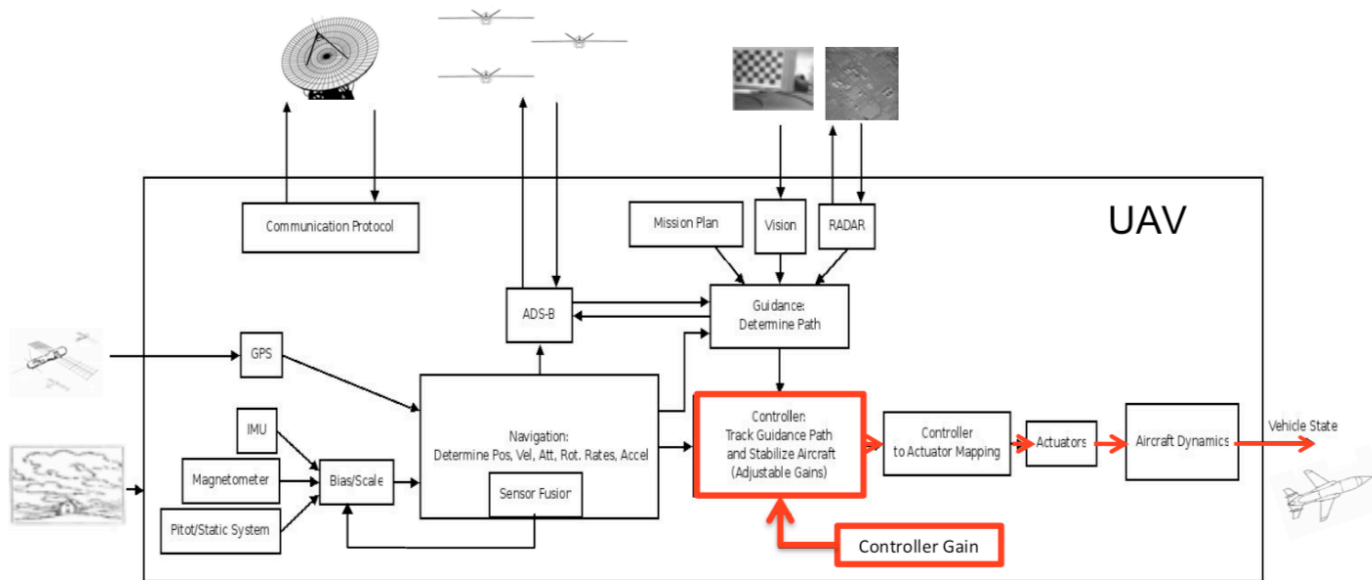


Figure 6: Flow of corruption due to gain scheduling attack.

Figure 6 shows the components that can be affected by the gain scheduling attack. This attack is an example of a control system security vulnerability. Gain scheduling is often used to control non-linear systems. For example, a UAV will need different gains for control depending on the state of the UAV (mass, altitude, speed, aps down, etc.). A UAV will have different dynamical properties depending on its state and will require gains matched to each state in order to control the vehicle properly. Gain scheduling is also used in hybrid systems. In hybrid systems, a system is assumed to have multiple modes of operation, and the modes can change at any given time following some rules. In the case of a UAV, for example, there might be different modes corresponding to take off, landing, and cruising. Each of these modes will have different gains for controlling the vehicle.

The control gains are often pre-computed and trusted, and they are coded into the on-board autopilots. Without strict monitoring of the software, an override of these gains could very well go undetected. Changes to the gains or gain scheduling logic could cause decreased performance in the autopilot or dangerous instability in a UAV. Additionally, even if all of the individual modes in a hybrid system are stable, some switching sequence could result in the system becoming unstable. This possibility could be used against the vehicle by an attacker. An attacker could also force infinite switching between gains, which will cause loss of control over a UAV.

Some of the possible attack methods are sensor spoofing to cause mode confusion, overriding gains through hacking, and causing denial of service between the controller gain block and the UAV controller block by overloading the on-board processor.

VI.B.2. Actuator/Sensor

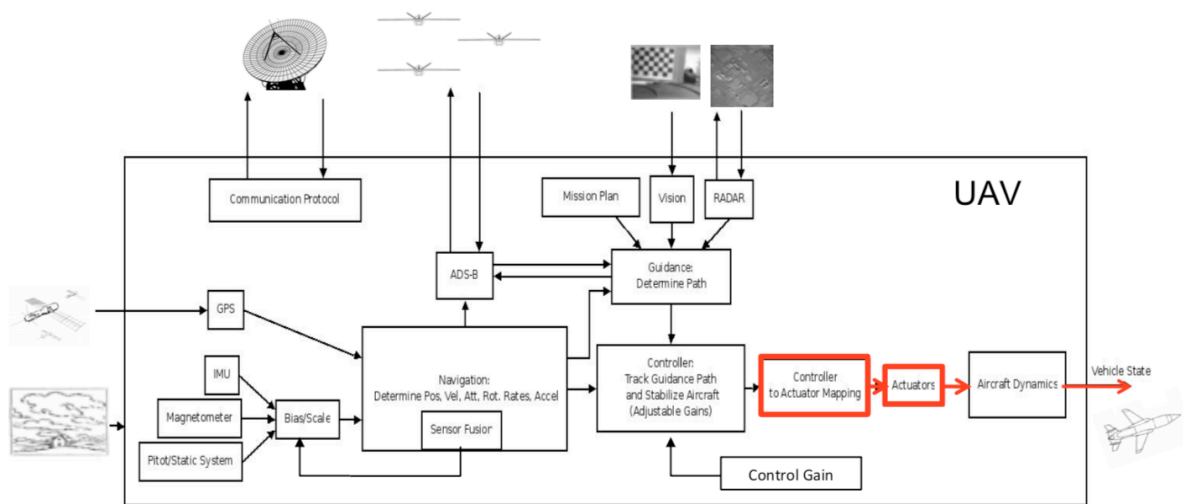


Figure 7: Flow of corruption due to actuator/sensor attack.

Figure 7 shows the components that can be affected by the actuator/sensor attack. This attack is an example of a control system security vulnerability. One of the easiest ways for corrupted data to affect the UAV control will be by affecting the control surface actuators and accompanying sensors. The UAV autopilot sends out commands to the actuators in order to modify the control surface state, which will modify the evolution of the vehicle state. If the commands going to the actuators were somehow overridden, the result would be the loss of control of the vehicle. Modifying the measurements of the actuator sensors in the systems that use them will also affect the state of the actuator and, thus, the state of the vehicle. This attack can result in loss of control of the UAV and instability in the UAV.

Some of the possible attack methods are overriding actuator state estimates, actuator mappings, or sensor readings; false data injection; and denial of service between the actuator and the controller interface by overloading the on-board processor. These attacks can be detected using a fault detection approach, but the algorithms currently being used are not robust enough or are too computationally costly. [10]

VI.B.3. GPS

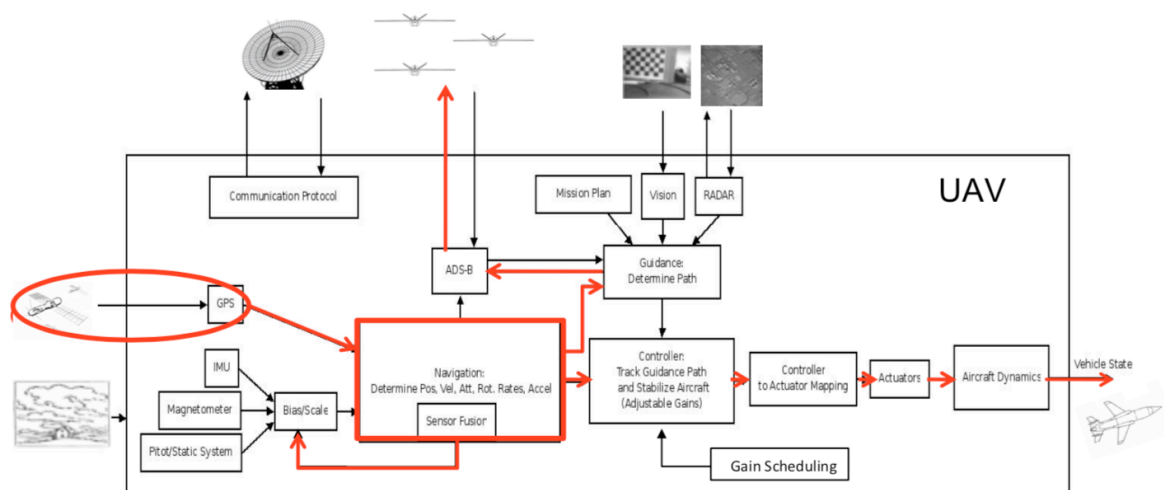


Figure 8: Flow of corruption due to GPS attack.

Figure 8 shows the components that can be affected by the GPS attack. This attack is an example of an application logic security vulnerability. Today, most UAV systems rely heavily on GPS data to locate themselves, the ground station, and their targets. The data received through the GPS sensors can be spoofed, which results in a false estimate of the UAV position in the on-board navigation system. If the UAV is fully automated, the on-board guidance system will then lead the UAV to a false target location or ground station if returning home. This type of attack will result in failed missions and possible loss of assets.

VI.B.4. Automatic Dependent Surveillance - Broadcast (ADS-B)

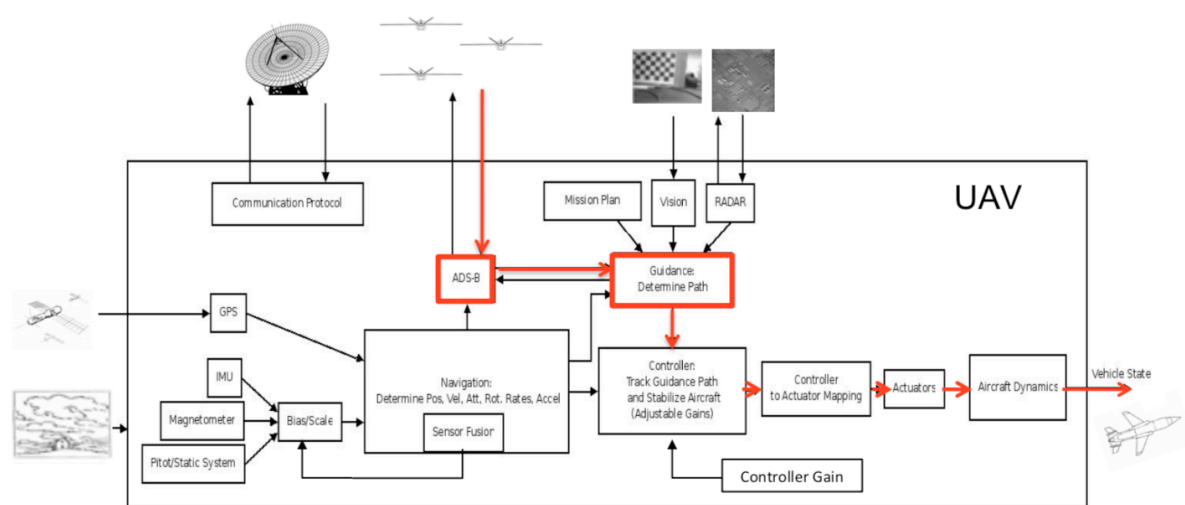


Figure 9: Flow of corruption due to ADS-B attack.

Figure 9 shows the components that can be affected by the ADS-B attack. This attack is an example of an application logic security vulnerability. ADS-B is an on-board component part of the next generation air traffic control system, which broadcasts information about an aircraft, such as position, heading, speed, and intent. For a UAV, this system will mainly be used for environmental awareness and collision avoidance, which is part of the navigation component of the autopilot. Since ADS-B is a broadcast system intended for all nearby aircraft, the data transmitted is not encrypted.

This creates an easy attack point for false data injection. The ADS-B data is used for navigation by the UAV autopilot, and false ADS-B data can accordingly throw the UAV off track during a mission. Also, if the ADS-B data is unavailable while another aircraft is en route for collision, the survivability of the UAV is affected greatly.

Some of the possible attack methods are spoofing ADS-B data and jamming. A multilateration verification method can be used to detect simple false broadcasts of the ADS-B data. This method uses two ADS-B sensors to estimate the direction of the broadcast source by measuring the time difference between the reception of the broadcasts on the sensors. Even with this verification, spoofing is harder to detect if the attacker is knowledgeable about the validation techniques. Figure 10 shows an example of a way to beat the multilateration verification method. [11] $\hat{\Psi}$ and $\hat{\Phi}$ represent the estimated direction of the ADS-B signal and the Aircraft (from the ADS-B data) respectively. It is clear that an aircraft that has zero conflict with another aircraft can spoof an ADS-B signal in order to fake an expected collision course, which will cause the other aircraft to change its course.

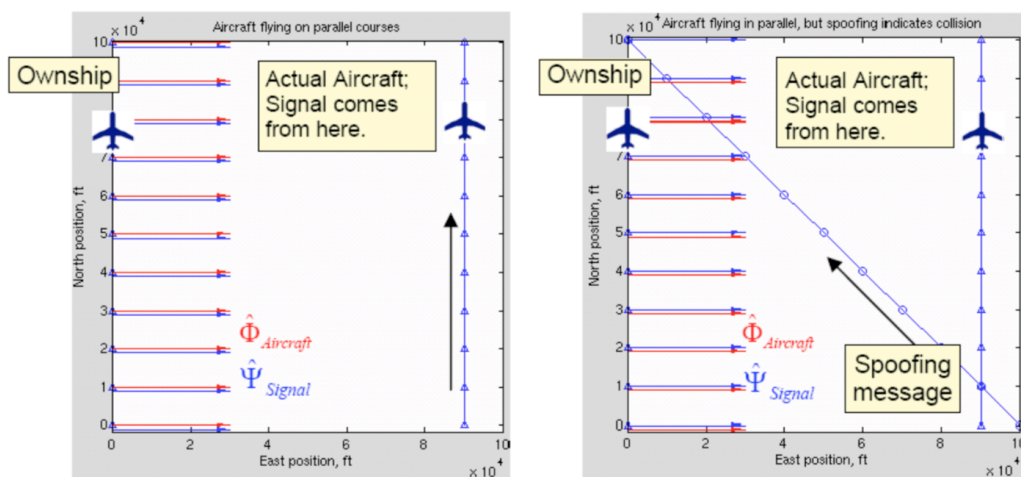


Figure 10: Example of failure of multilateration verification of ADS-B signal.

VI.B.5. Fuzzing

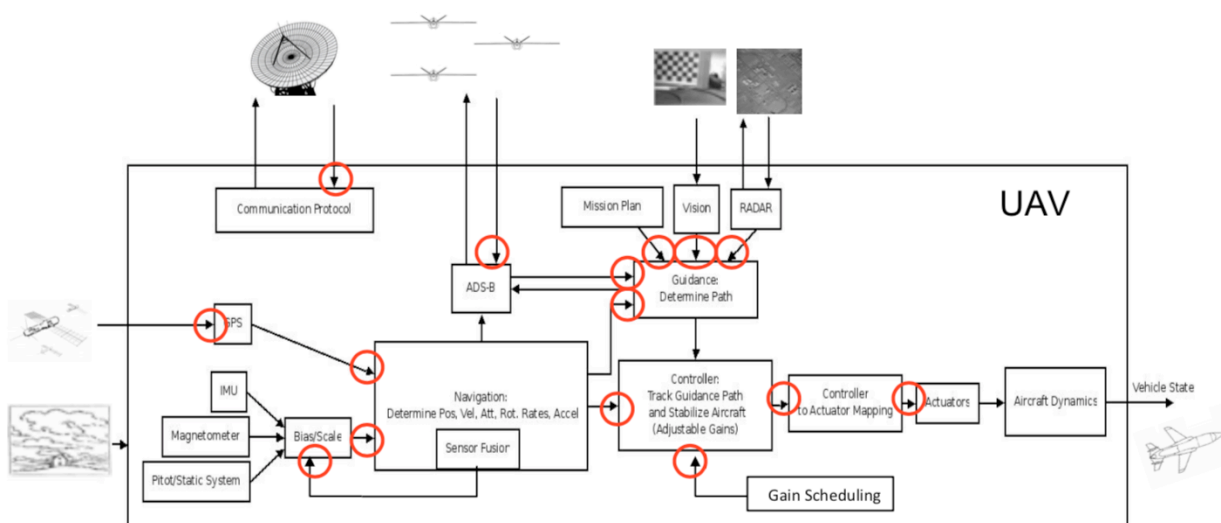


Figure 11: Attack points of fuzzing attack.

Figure 11 shows the attack point of the fuzzing attack. This attack is an example of a control system security and application logic security vulnerability. The concept of software fuzzing can be applied to GNC algorithms. In the UAV autopilot system, random inputs with expected distributions are not uncommon, and Gaussian noise inputs are routinely accounted for. However, unexpected, invalid, or completely random inputs can cause unknown behavior. If an attacker can somehow access any of the data flow between components and corrupt them with junk values, it will cause unexpected problems for the autopilot system. The consequences for this type of attack could include aircraft instability, process lock-up, and invalid outputs to the next process.

Some of the possible attack methods are buffer overflow attacks, sending malicious packets with invalid payload data to the UAV, and adding malicious hardware between components. It is also possible to use malicious fuzzing as a tool to discover vulnerabilities. Intentional white-box and black-box fuzzing tests could be performed on the system to determine the robustness of the GNC algorithms. [12]

VI.B.6. Digital Update Rate

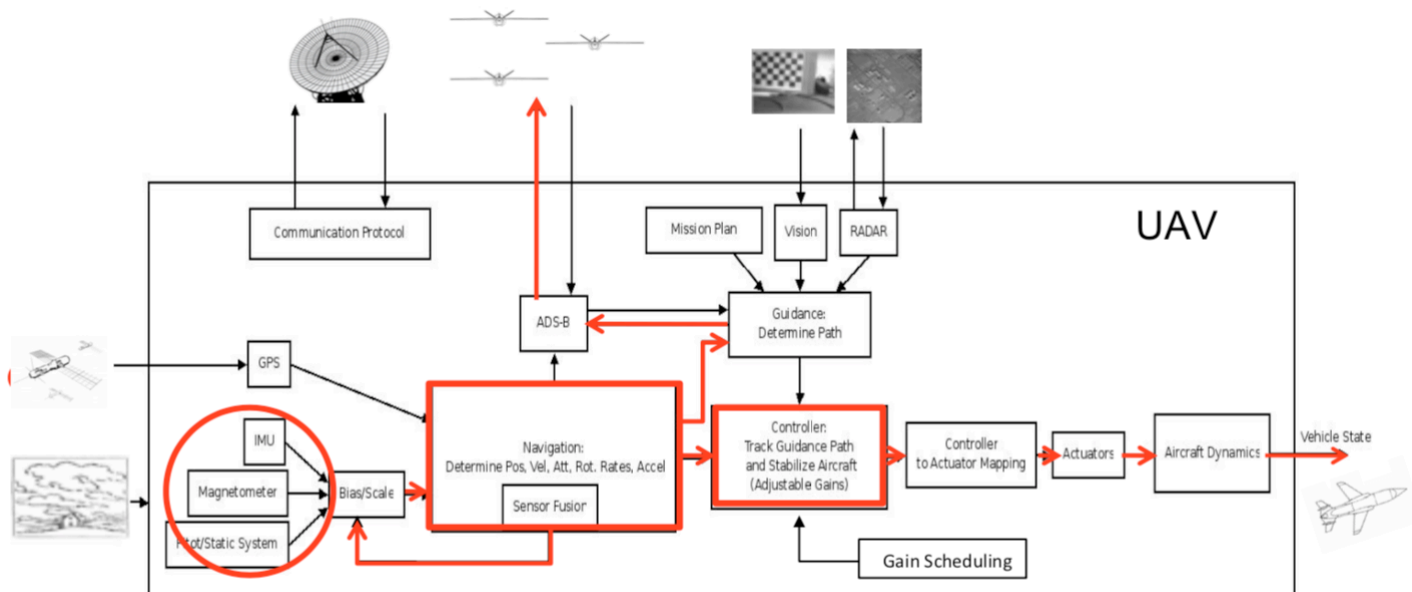


Figure 12: Flow of corruption due to digital update rate attack.

Figure 12 shows the components that can be affected by the digital update rate attack. This attack is an example of a control system security and application logic security vulnerability. UAV autopilots are digital computers and, accordingly, any inputs or outputs of the autopilot are discretized. This means that any continuous inputs to the autopilots are converted to digital inputs through discrete sampling. If the autopilot was designed with a continuous controller, that controller is also converted to a discretized form. For a discretized system, as the sample time increases, the system becomes unstable/uncontrollable. [13] For data collection, longer sampling periods will increase the probability of data aliasing, as shown in Figure 13.

Some of the possible attack methods are changing the sampling time of analog-to-digital converters through buffer overflow or hardware manipulation and denial of service attacks that prevent the processor from running the controller or navigator at the desired update rate.

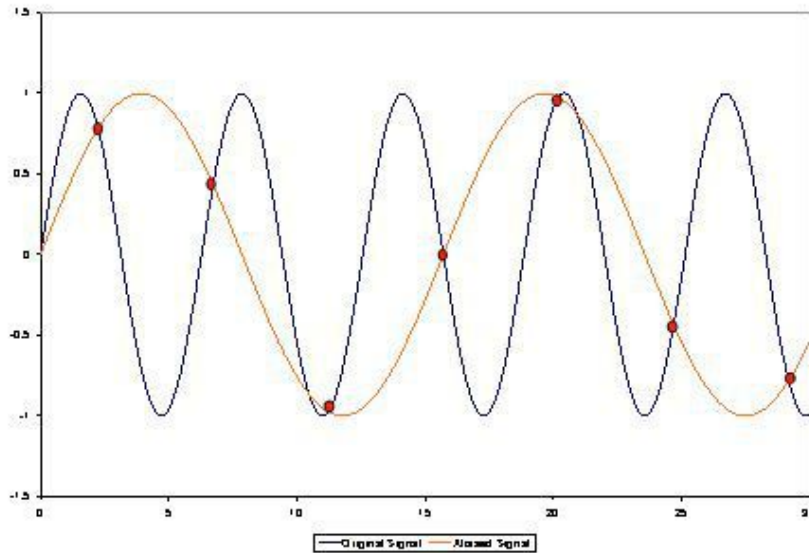


Figure 13: Example of aliasing of data due to insufficient sampling rate.

VII. Post-Attack Behavior Analysis

VII.A. Simulation Tools

VII.A.1. ScicosLab and Scicos

ScicosLab is a testbed that is very similar to MatLab and Simulink, but is an open source alternative. Like MatLab, ScicosLab allows easy matrix operations and the ability to execute mathematical functions and scripts. Scicos, like Simulink, is a block-based graphical simulation suite designed for control system design and simulation. We use Scicos to run ScicosLab, C, and C++ code underlying the graphical form. Since numerical analysis of aircraft dynamics requires large mathematical calculations, ScicosLab and Scicos are used regularly in our lab to carry out these simulations.

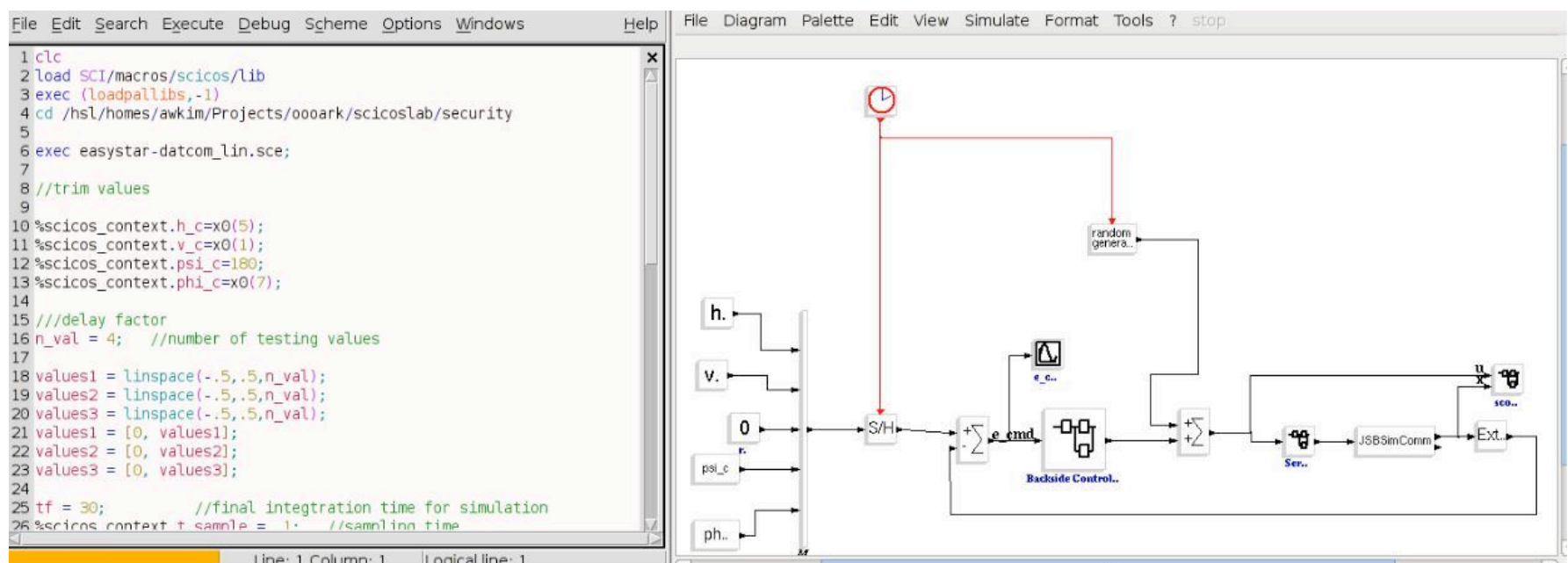


Figure 14: Screen shot of ScicosLab interface on the left and Scicos interface on the right.

VII.A.2. Arkscicos

Arkscicos (Autonomous Robotics Kit Toolbox for Scicos) is a software library for modeling and simulating unmanned flight developed by the Purdue Hybrid Systems Lab. Arkscicos was developed to be used with ScicosLab and Scicos. This software library features:

- Hardware In the Loop (HIL) interfaces
- Sensor models including Global Positioning System (GPS), Inertial Navigation System (INS), GPS/INS, Vision, and Magnetometer
- Visualizations Data logging
- High fidelity 6DOF aircraft dynamics

VII.A.3. JSBSim

JSBSim is a numerical analysis tool that simulates high-fidelity, 6-degree-of-freedom aircraft dynamics. The relevant equations of motion are derived by Stevens and Lewis [13]. JSBSim has a large library of aircraft models available, and can load custom models. JSBSim also has the ability to incorporate weather conditions into its numerical analysis.

Our lab has a small, easy to fly R/C airplane called the Easy Star that has been adapted for use as a UAV. The Easy Star uses the rudder and elevator as control surfaces. The rudder, elevator, and throttle comprise the three inputs into the system. Our custom autopilot is used on the Easy Star, so we have developed an accurate JSBSim model of it to use in developing and testing the autopilot before live flight. The Easy Star is shown in Figure 15 being tested in the wind tunnel at Purdue. We will use the Easy Star model to test our identified cyber threats and vulnerabilities of the current UAV autopilot system.



Figure 15: Picture of the Easy Star being tested in the Boeing wind tunnel at Purdue.

VII.A.4. Flight Gear

Flight Gear is an open source flight simulator we use to visualize numerical simulations. Flight Gear can receive data from JSBSim and display the physical actions of an aircraft accurately in a simulated real world environment. Figure 16 demonstrates a simulation of a plane flying through San Francisco.



Figure 16: Screen shot of a flight simulation on Flight Gear.

VII.A.5. Purdue HSL Analysis Tool

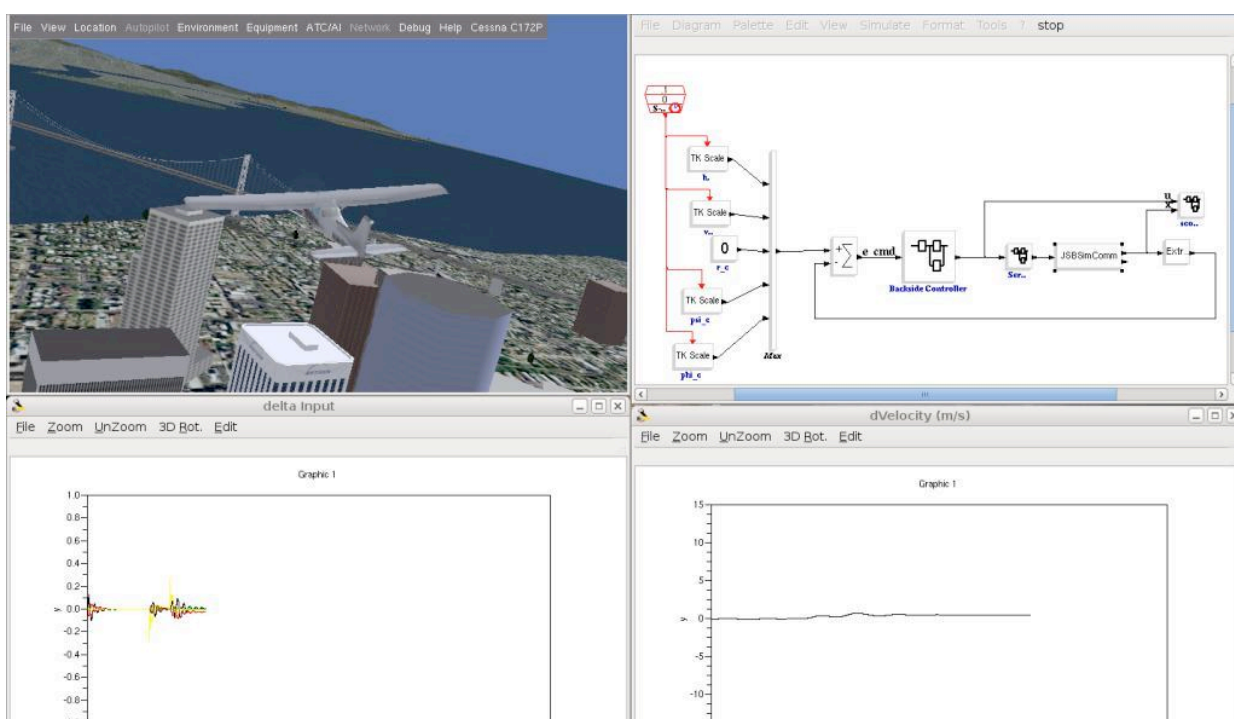


Figure 17: Screen shot of HSL Analysis Tool being used.

At the Purdue Hybrid Systems Lab (HSL), we have developed a high-fidelity simulation testbed for fully-autonomous unmanned aircraft using all of the aforementioned tools. We can study and test many aspects of the unmanned aircraft, such as command and control, sensors, and stability. Figure 17 shows a simulation being run using the Purdue HSL Analysis Tool, which can log any data while visually simulating the aircraft motion. We have also helped to develop a ground station for fully autonomous UAVs, which can be seen in Figure 18. Again, the motion of the UAV is visually stimulated while all the data of the simulation can be logged. The ground control provides a display of UAV

information such as position, speed, heading, and sensor data, along with the control over the UAV. With our Purdue HSL Analysis Tool, we performed numerical analysis on the effects of the identified cyber attacks on the UAV autopilot system.

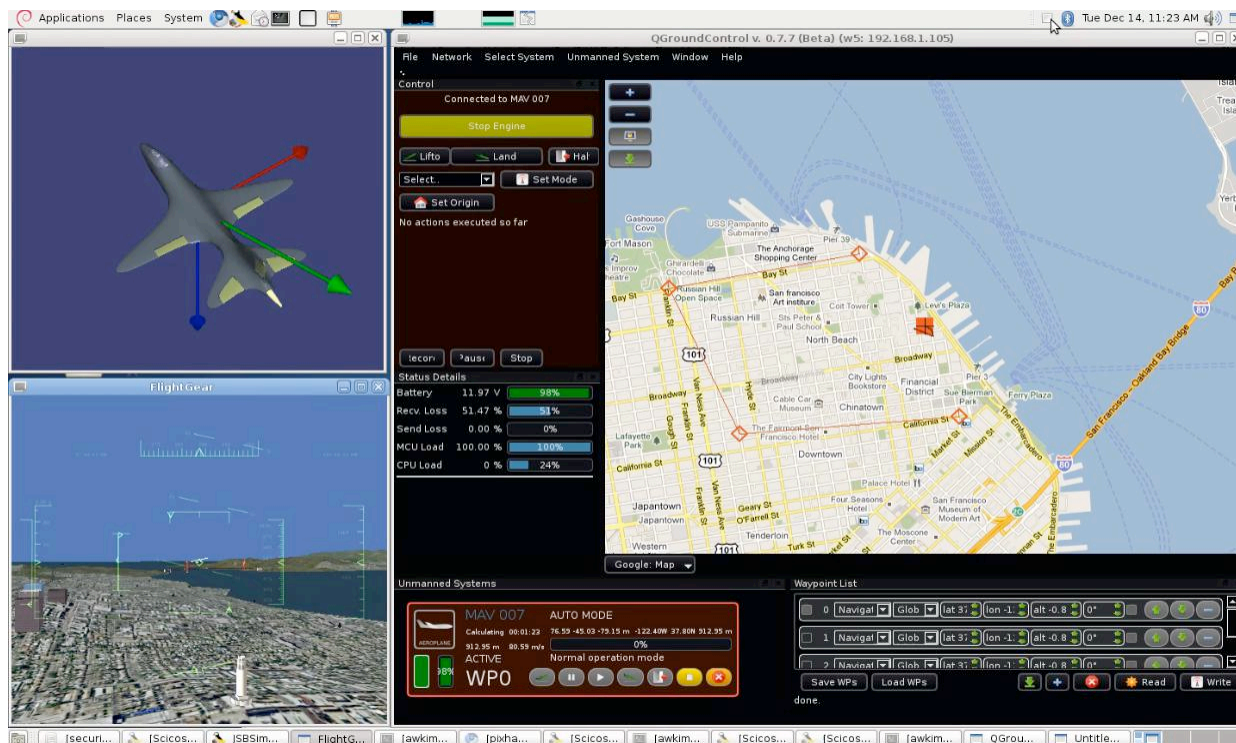


Figure 18: Screen shot of the ground control software (QGroundControl) and flight simulation.

VII.B. Gain Scheduling

VII.B.1. Analysis

The gain scheduling attack can be used in several different ways as explained previously. For this study, we have chosen to simulate the case where the UAV being attacked has several pre-programmed trim state stabilization gains for different flight conditions and the attacker causes a switch to a set of gains that does not match the current flight conditions. The autopilot is used mostly during the long period cruise portion of a mission, which gives the attacker a big window for an opportunity to use the gain scheduling attack.

The simulation was run on a flight of the Easy Star UAV (Figure 15) model in a trim cruise condition of:

- Altitude: 1000 ft
- Speed: 45 ft/s
- Heading: 180 deg
- Bank Angle: 0 deg

Five values were measured during the UAV cruise: the deviations from the trim values of the altitude, speed, heading, bank angle, and the pitch angle. Measuring these deviations shows the stability of the UAV under the autopilot control. The gain scheduling attack was simulated by flying the Easy Star at a flight condition far removed from trim. This was

done to confirm the danger of operating a UAV with the wrong set of gains. The measured deviations and the visualization of the flight clearly show that this attack is detrimental to the stability of the flight compared to the normal case, confirming our hypothesis. A typical response of the UAV is plotted in Figure 19.

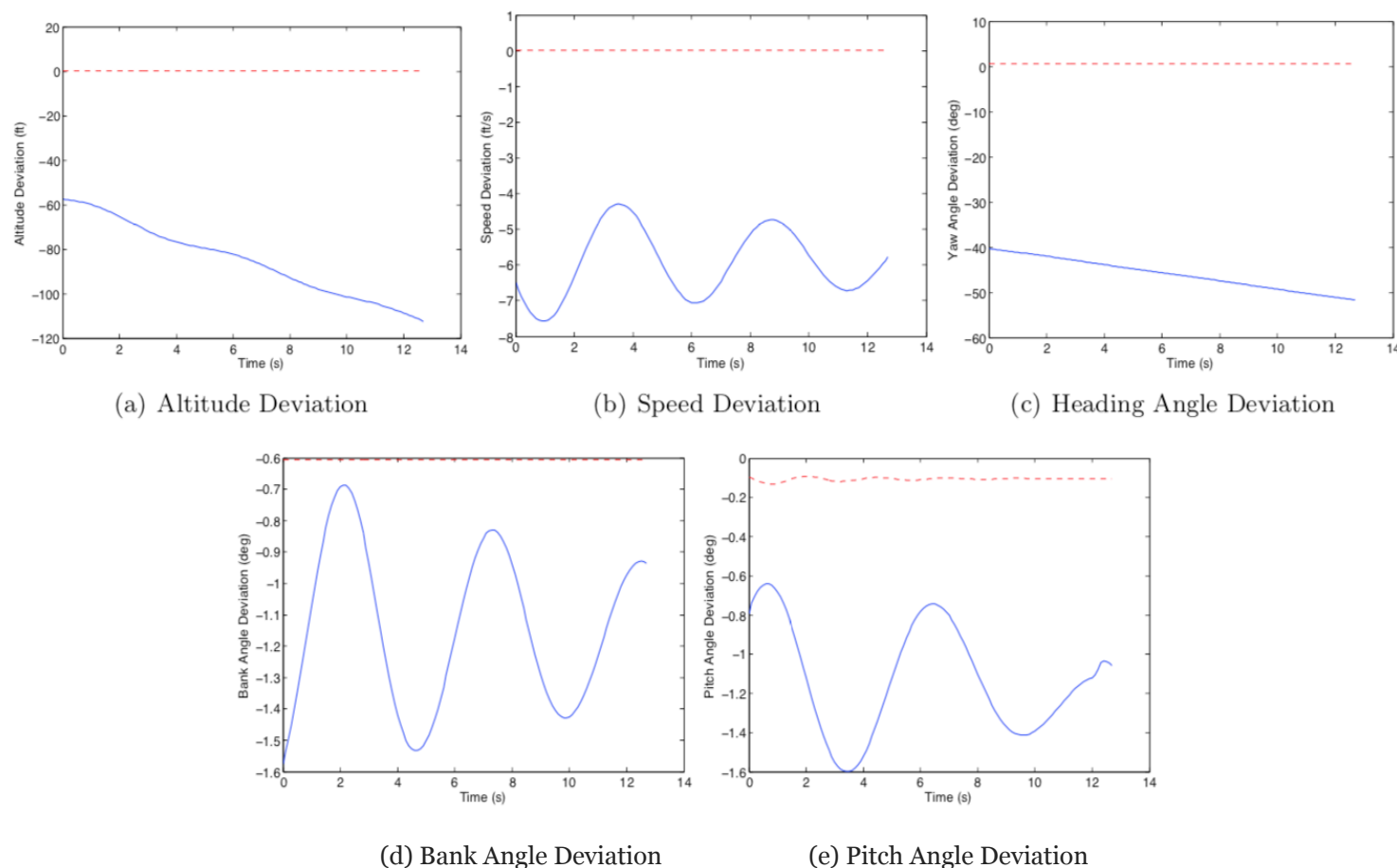


Figure 19: Sample data for the gain scheduling attack. The red dotted line represents the normal case and the blue solid line represents the case of an attack. This sample data was collected after the start of the simulation. As such, zero time here represents the start of the data collection and not the start of the simulation.

VII.B.2. Sensitivity Study

In order to identify the most effective use of the gain scheduling attack, a sensitivity study was performed by simulating the flight of Easy Star at varying flight conditions and keeping the constant control gains. Each of the flight conditions (altitude, speed, bank angle) were varied to generate 125 data points. The following shows the values used:

- **Flight Altitude:** five evenly spaced values between -50% and 50% offset from normal trim, and normal case
- **Flight Speed:** five evenly spaced values between -50% and 50% offset from normal trim, and normal case
- **Bank Angle:** five evenly spaced values between -6° and 6° offset from trim, and normal case

Five values were measured during the UAV flight, which are the squares of the deviations from the trim values of the altitude, speed, heading, bank angle, and the pitch angle. These values were plotted to identify which flight stability factor is most sensitive to which flight condition change (gain set change). This was done by creating plots in the following way:

1. Pick a design variable (in our case, flight altitude, flight speed, or bank angle).
2. Create a plot where the x-axis will represent the chosen design variable.
3. Choose a measurement variable (in our case, squares of the deviations from the trim values of the altitude, speed, heading, bank angle, and the pitch angle), which will be plotted on the y-axis against the chosen design variable.
4. Plot all the data that corresponds to each of the design variable values.
5. Calculate the means of the data set that corresponds to each of the design variable values and plot them.
6. Observe the trend of the mean values of the measurement data due to the design variable. If a linear trend line is used, the steeper slope of the trend line indicates higher sensitivity of the measurement variable to the design variable.

Figures 20, 21, 22, 23, and 24 show the sensitivity plot for altitude, speed, heading, bank angle, and the pitch angle respectively. By observing each of the plots, we concluded that for each stability factor, the following gain change attacks are most effective:

- Altitude Stability: Gain change for lower **altitude**
- Speed Stability: Gain change for higher **bank angle**
- Yaw Stability: Gain change for different operating **speed**
- Roll Stability: Not one attack point is more vulnerable
- Pitch Stability: Gain change for lower **bank angle**

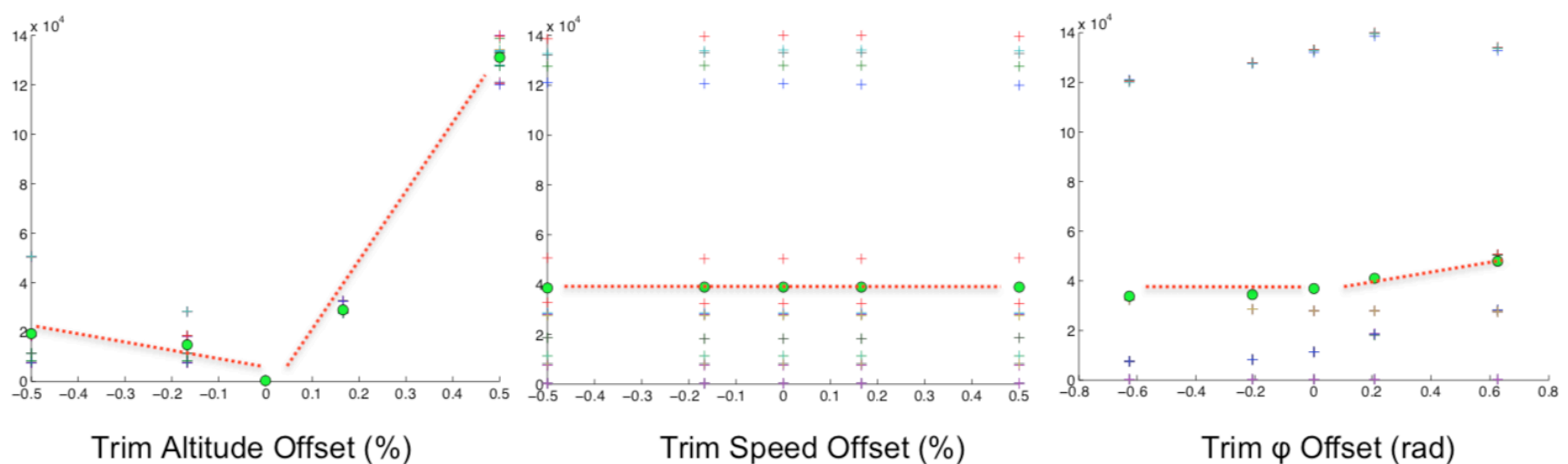


Figure 20: Sensitivity comparison plots for gain scheduling attack effect on altitude stability. The y-axis represents the square of deviation of altitude from trim condition (ft^2). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear trend line of the mean values.

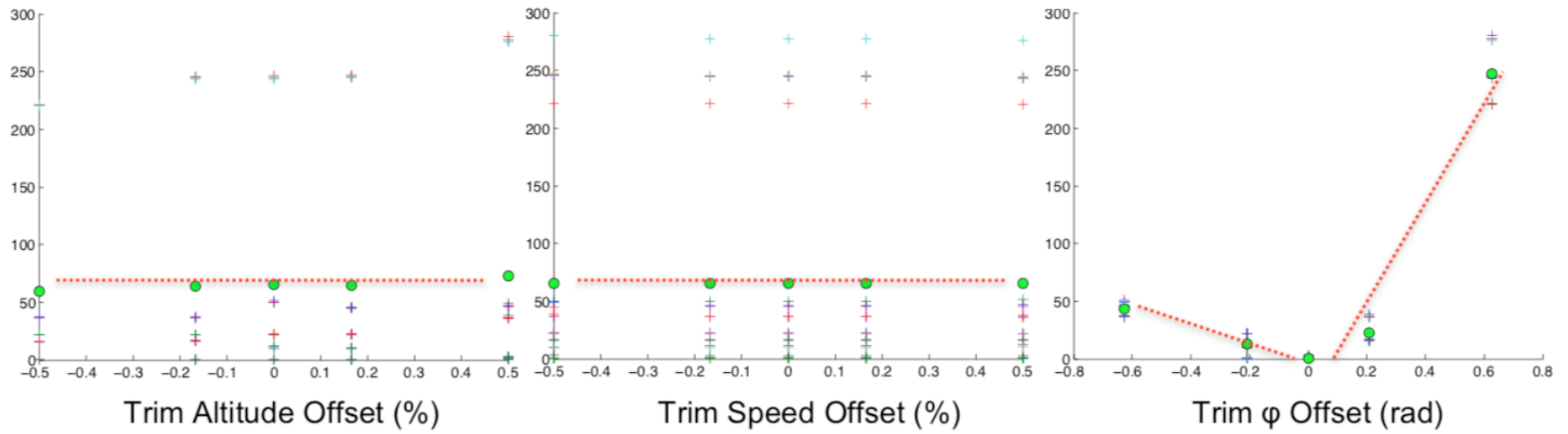


Figure 21: Sensitivity comparison plots for gain scheduling attack effect on speed stability. The y-axis represents the square of deviation of speed from trim condition ($f t^2 = s^2$). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear fit line of the mean values.

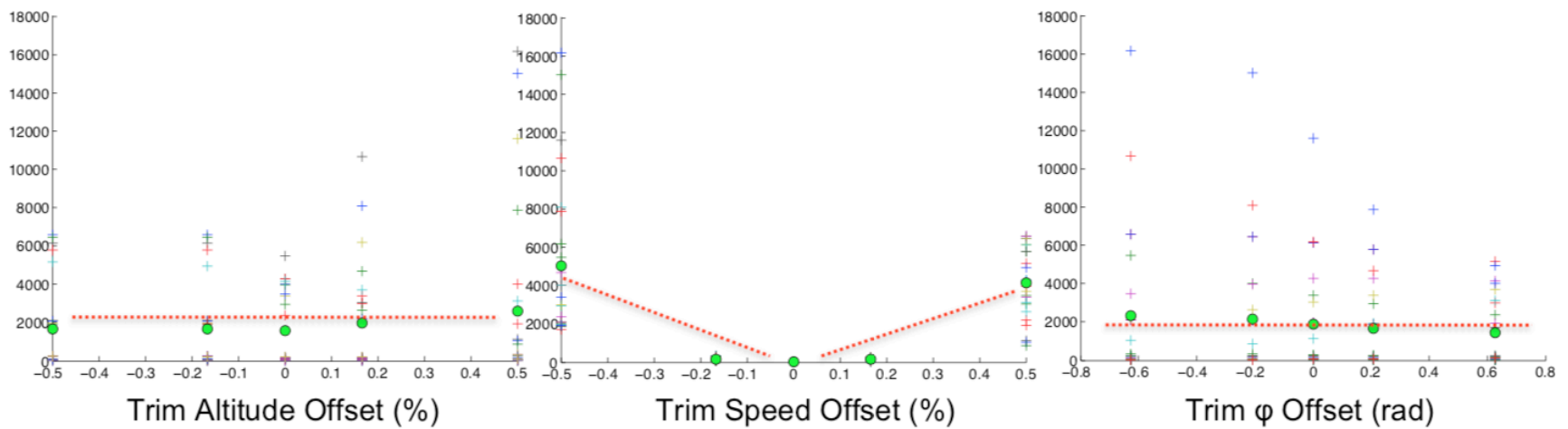


Figure 22: Sensitivity comparison plots for gain scheduling attack effect on heading stability. The y-axis represents the square of deviation of heading from trim condition (deg^2). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear fit line of the mean values.

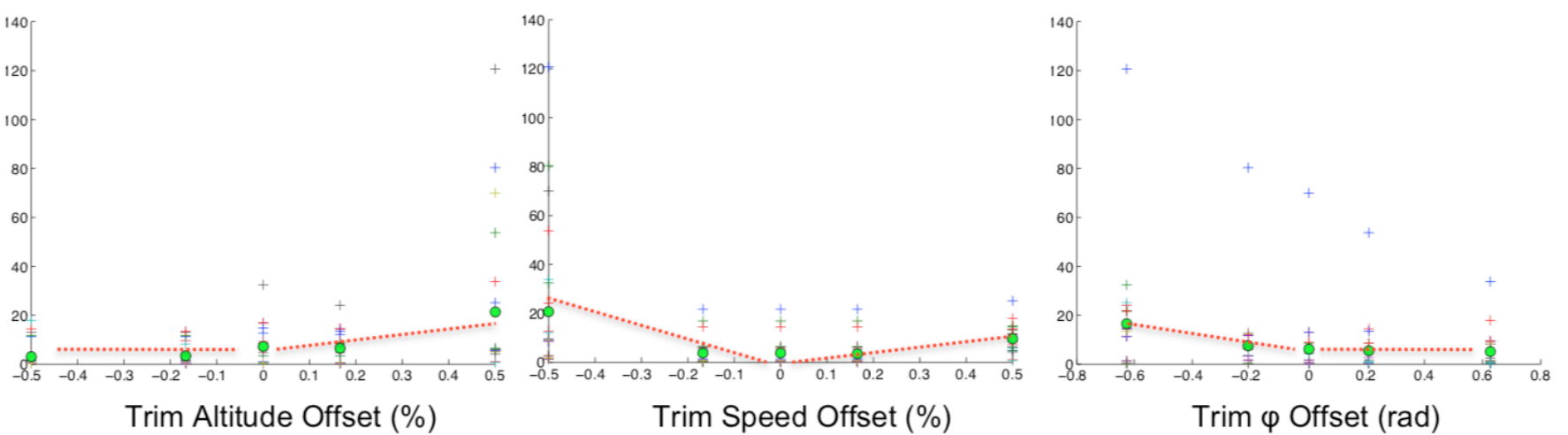


Figure 23: Sensitivity comparison plots for gain scheduling attack effect on bank angle stability. The y-axis represents the square of deviation of bank angle from trim condition (deg^2). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear fit line of the mean values.

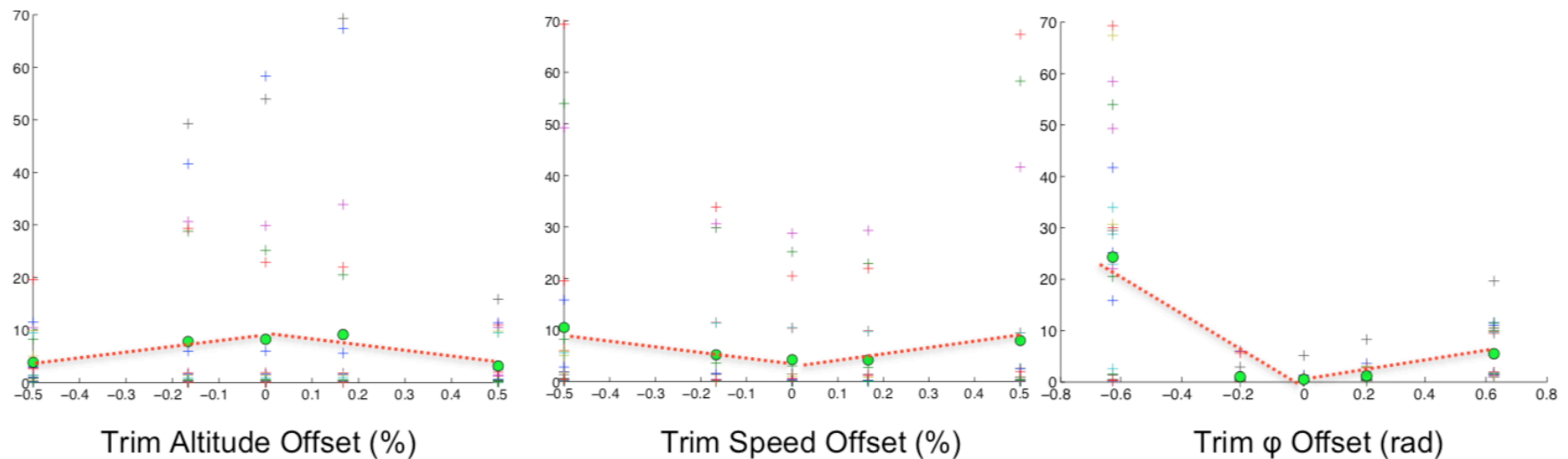


Figure 24: Sensitivity comparison plots for gain scheduling attack effect on pitch angle stability. The y-axis represents the square of deviation of pitch angle from trim condition (deg^2). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear t line of the mean values.

VII.C. Actuator/Sensor

We have carried out the numerical analysis of the fuzzing attack on the actuators, which is precisely the actuator/sensor attack. See the fuzzing section for the results.

VII.D. GPS

The study into the GPS vulnerabilities of autopilot systems proved to be a large undertaking in and of itself. We are currently carrying out research into this subject matter, which is out of the scope of this paper. We do report that the GPS spoofing can lead a UAV astray and with proper techniques, the attack is undetectable while giving limited control over the destination of the UAV to the attacker.

VII.E. Fuzzing

VII.E.1. Analysis

The fuzzing attack can be injected at several different points in the autopilot data flow. For this study we have chosen to simulate the case where the attacker injects random inputs to the actuators (i.e. corrupt the data from the controller to the actuators). This was done to also simulate the actuator attack. The simulation was run on a flight of the Easy Star UAV (Figure 15) model in a trim cruise condition of:

- Altitude: 1000 ft
- Speed: 45 ft/s
- Heading: 180 deg
- Bank Angle: 0 deg

Five values were measured during the UAV cruise, which are the deviations from the trim values of the altitude, speed, heading, bank angle, and the pitch angle. Measuring these deviations shows the stability of the UAV under the autopilot control. The fuzzing attack was simulated by injecting random Gaussian noise into each of the actuator inputs (throttle, elevator, rudder) at a 10 Hz rate. This was done to confirm the danger of operating a UAV under attack. The measured deviations and the visualization of the flight clearly show that this attack affects the stability of the flight compared to the normal case, confirming our hypothesis. A typical response of the UAV is plotted in Figure 25.

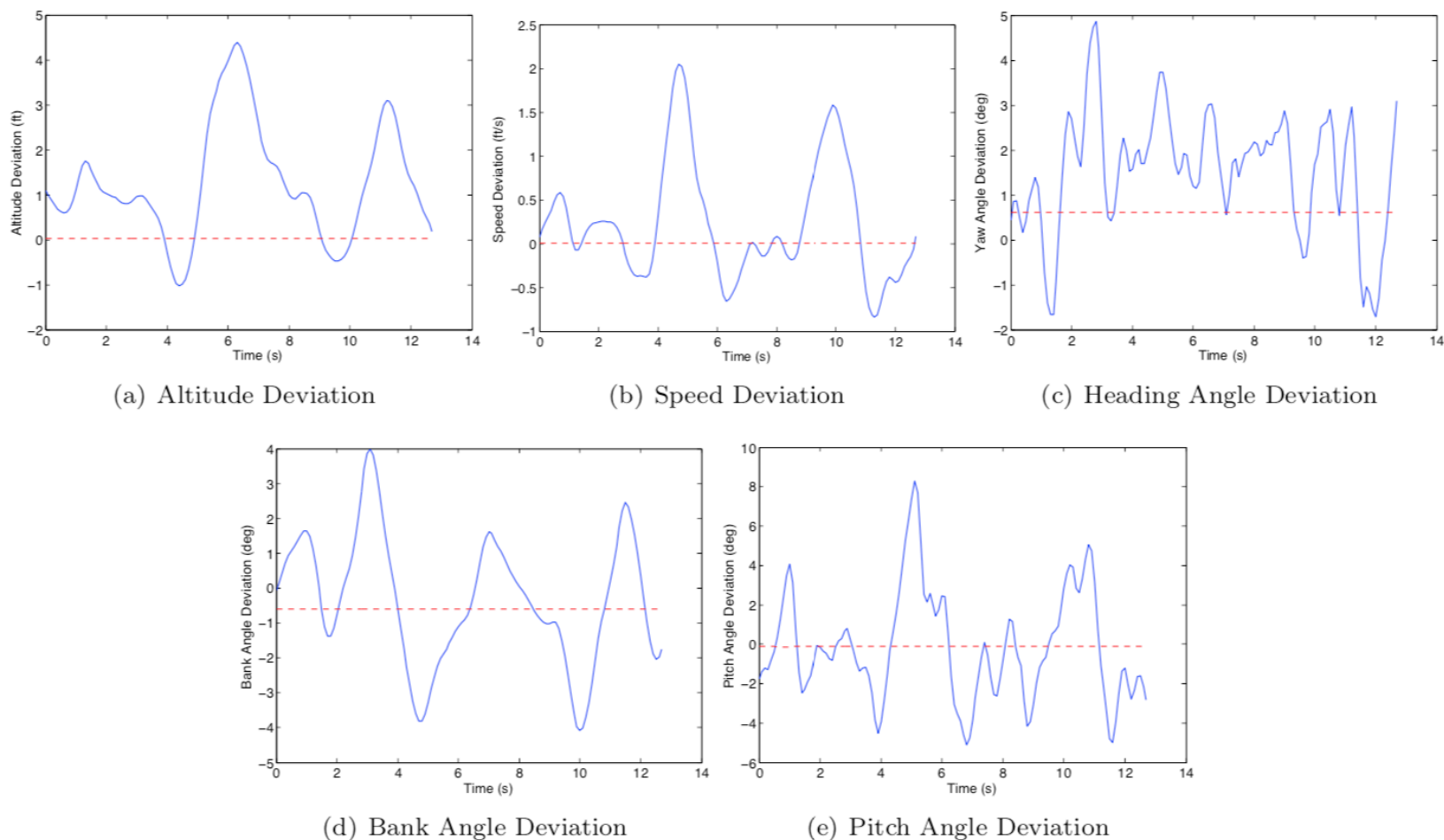


Figure 25: Sample data for the fuzzing attack. The red dotted line represents the normal case and the blue solid line represents the case of an attack. This sample data was collected after the start of the simulation. As such, zero time here represents the start of the data collection and not the start of the simulation.

VII.E.2. Sensitivity Study

In order to identify the most effective noise injection point, a sensitivity study was performed by simulating the flight of Easy Star at the trim condition and injecting Gaussian noise with varying standard deviations to each of the actuators. The following shows the values used to create 125 data points:

- Throttle: five evenly spaced values from 0 to 0.2
- Elevator: five evenly spaced values from 0 to 0.2
- Rudder: five evenly spaced values from 0 to 0.2

Since this simulation was stochastic, Monte Carlo simulations of 10 runs were run for each of the test cases. Five values were measured during the UAV flight, which are the average of the squares of the deviations from the trim values of the altitude, speed, heading, bank angle, and the pitch angle from the Monte Carlo runs. These values were plotted in the same manner as in the sensitivity study of the gain scheduling attack to identify which flight stability is most sensitive to which data corruption.

Figures 26, 27, 28, 29, and 30 show the sensitivity plot for altitude, speed, heading, bank angle, and the pitch angle respectively. By observing each of the plots, we concluded that for each of the stability factors, the following injection points are most effective:

- Altitude Stability: Rudder input noise
- Speed Stability: Rudder input noise
- Yaw Stability: Elevator input noise
- Roll Stability: Elevator input noise
- Pitch Stability: Rudder input noise

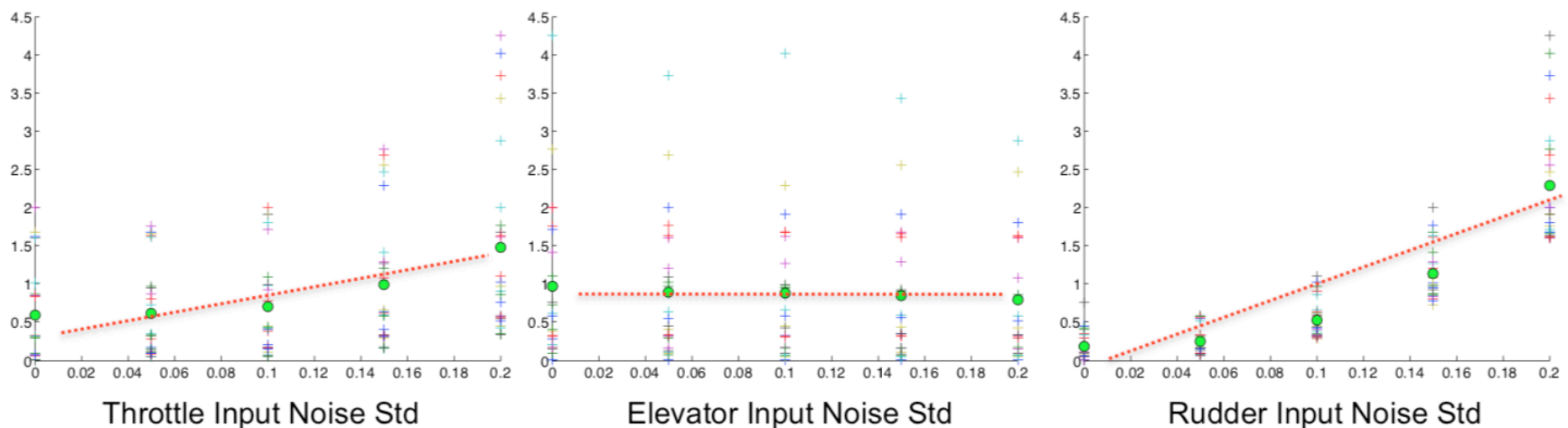


Figure 26: Sensitivity comparison plots for fuzzing attack effect on altitude stability. The y-axis represents the square of deviation of altitude from trim condition (ft^2). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear t line of the mean values.

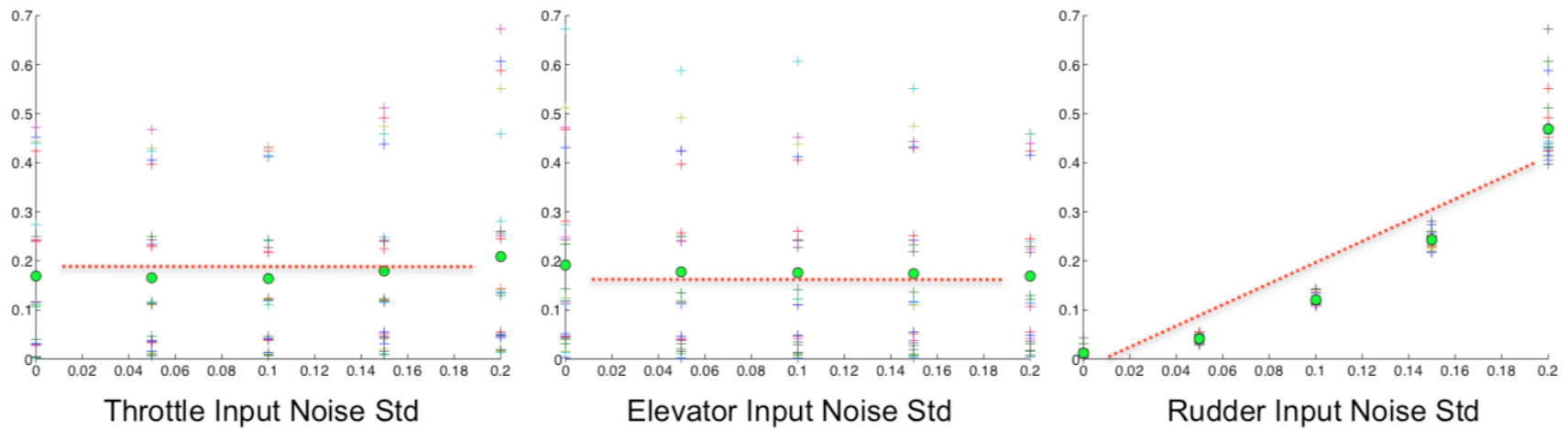


Figure 27: Sensitivity comparison plots for fuzzing attack effect on speed stability. The y-axis represents the square of deviation of speed from trim condition ($f t^2 = s^2$). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear t line of the mean values.

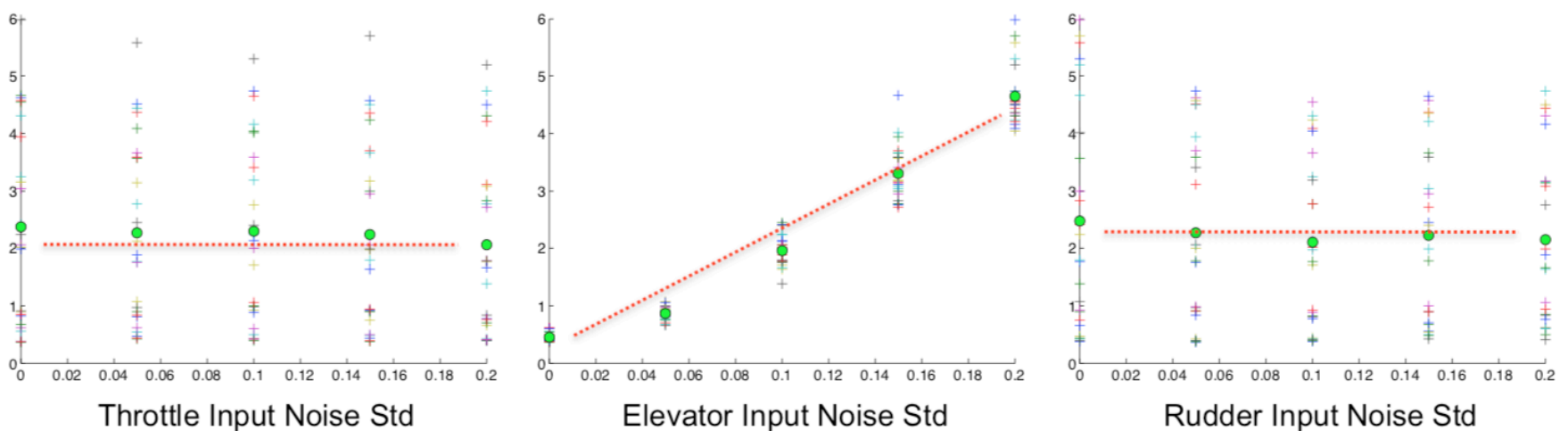


Figure 28: Sensitivity comparison plots for fuzzing attack effect on heading stability. The y-axis represents the square of deviation of heading from trim condition (deg^2). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear t line of the mean values.

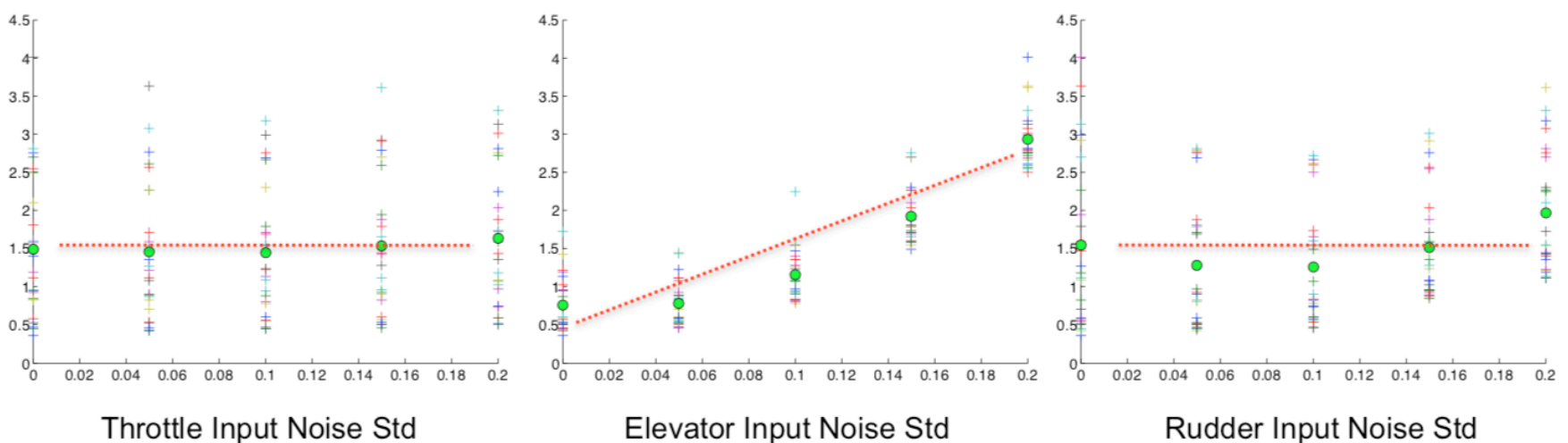


Figure 29: Sensitivity comparison plots for fuzzing effect on bank angle stability. The y-axis represents the square of deviation of bank angle from trim condition (deg^2). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear t line of the mean values.

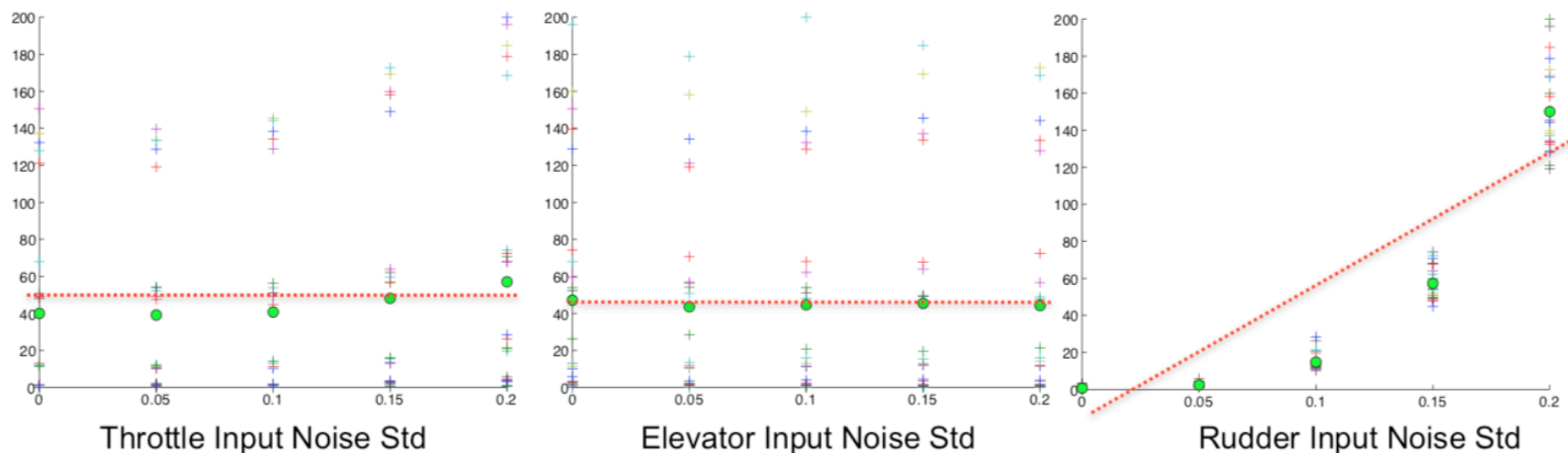


Figure 30: Sensitivity comparison plots for fuzzing effect on pitch angle stability. The y-axis represents the square of deviation of pitch angle from trim condition (deg^2). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear t line of the mean values.

VII.F. Digital Update Rate

VII.F.1. Analysis

The digital update rate attack can be injected at several different points in the autopilot data flow. For this study, we have chosen to simulate the case where the attacker delays inputs to the actuators (i.e. delay the data from the controller to the actuators).

The simulation was run on a flight of the Easy Star UAV (Figure 15) model in a trim cruise condition of:

- Altitude: 1000 ft
- Speed: 45 ft/s
- Heading: 180 deg
- Bank Angle: 0 deg

Five values were measured during the UAV cruise, which are the deviations from the trim values of the altitude, speed, heading, bank angle, and the pitch angle. Measuring these deviations shows the stability of the UAV under the autopilot control. The digital update rate attack was simulated by delaying the inputs from the controller to each of the actuators (throttle, elevator, rudder). This was done to confirm the danger of operating a UAV under attack. The measured deviations and the visual of the flight clearly show that this attack affects the stability of the flight compared to the normal case, confirming our hypothesis. A typical response of the UAV is plotted in Figure 31.

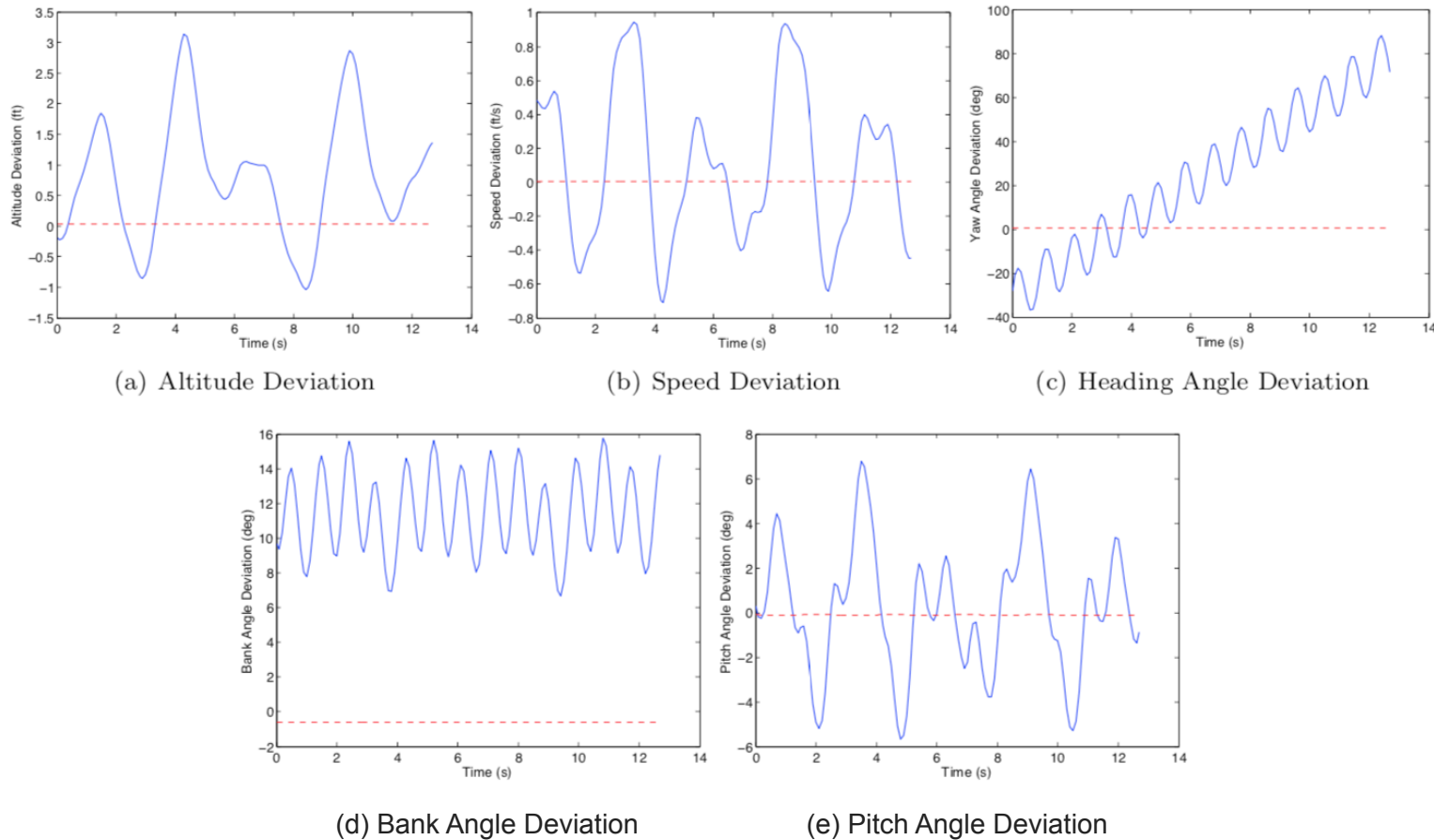


Figure 31: Sample data for the digital time update attack. The red dotted line represents the normal case and the blue solid line represents the case of an attack. This sample data was collected after the start of the simulation. As such, zero time here represents the start of the data collection and not the start of the simulation.

VII.F.2. Sensitivity Study

In order to identify the most effective data delay injection point, a sensitivity study was performed by simulating the flight of Easy Star at the trim condition and delaying the inputs to each of the actuators by varying times. The following shows the values used to create 125 data points:

- Throttle: five evenly spaced values from 0 to 0.2 seconds of delay
- Elevator: five evenly spaced values from 0 to 0.2 seconds of delay
- Rudder: five evenly spaced values from 0 to 0.2 seconds of delay

Five values were measured during the UAV flight, which are the squares of the deviations from the trim values of the altitude, speed, heading, bank angle, and the pitch angle. These values were plotted in the same manner as in the sensitivity study of the gain scheduling attack to identify which flight stability is most sensitive to which input delay.

Figures 32, 33, 34, 35, and 36 show the sensitivity plot for altitude, speed, heading, bank angle, and the pitch angle respectively. By observing each of the plots we concluded that for each of the stability factors, the following delays are most effective:

- Altitude Stability: Rudder input delay
- Speed Stability: Rudder input delay
- Yaw Stability: Rudder input noise
- Roll Stability: Elevator input delay
- Pitch Stability: Rudder input noise

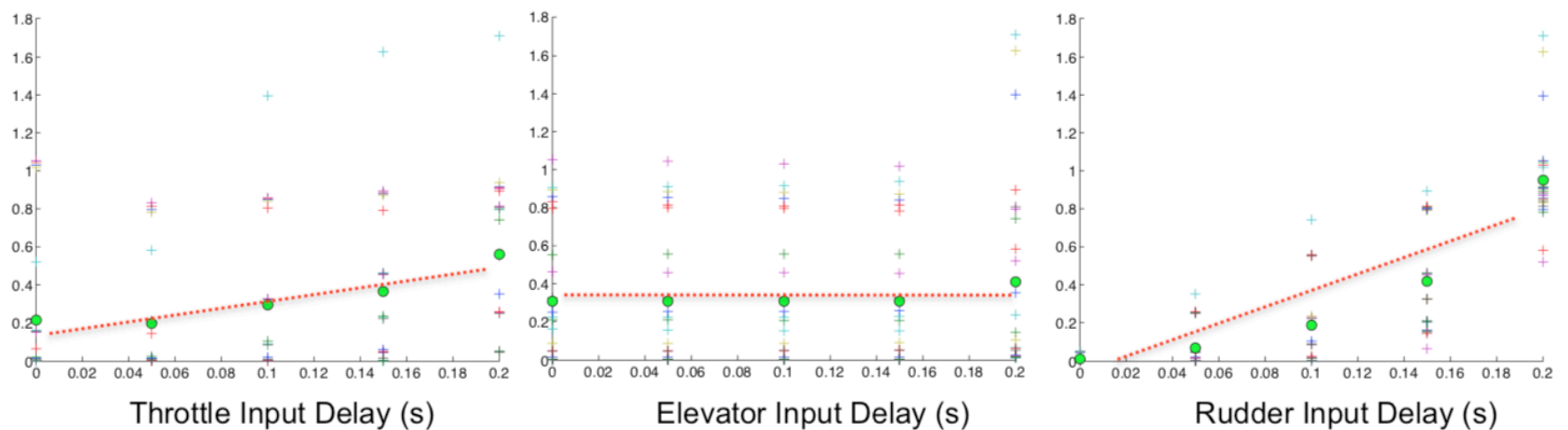


Figure 32: Sensitivity comparison plots for digital update rate attack effect on altitude stability. The y-axis represents the square of deviation of altitude from trim condition (ft^2). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear t line of the mean values.

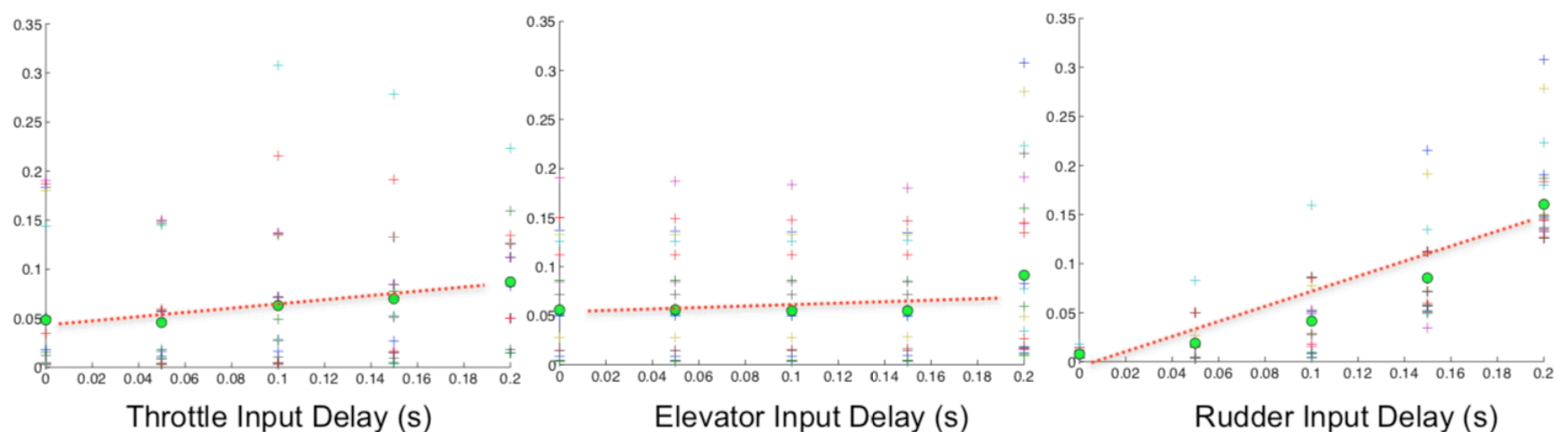


Figure 33: Sensitivity comparison plots for digital update rate attack effect on speed stability. The y-axis represents the square of deviation of speed from trim condition (ft^2/s^2). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear t line of the mean values.

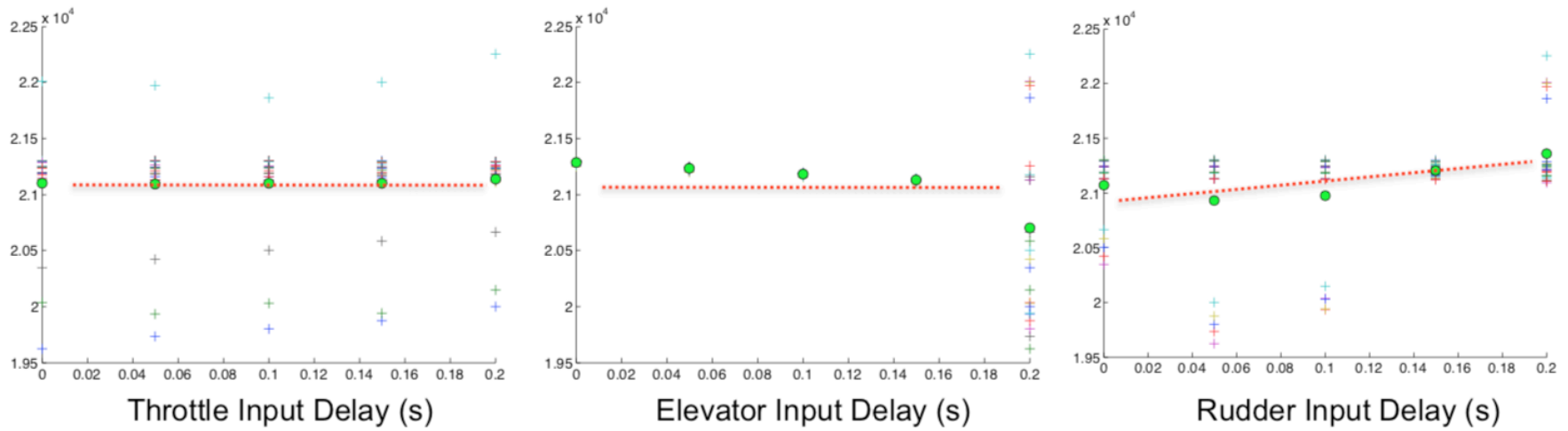


Figure 34: Sensitivity comparison plots for digital update rate attack effect on heading stability. The y-axis represents the square of deviation of heading from trim condition (deg^2). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear t line of the mean values.

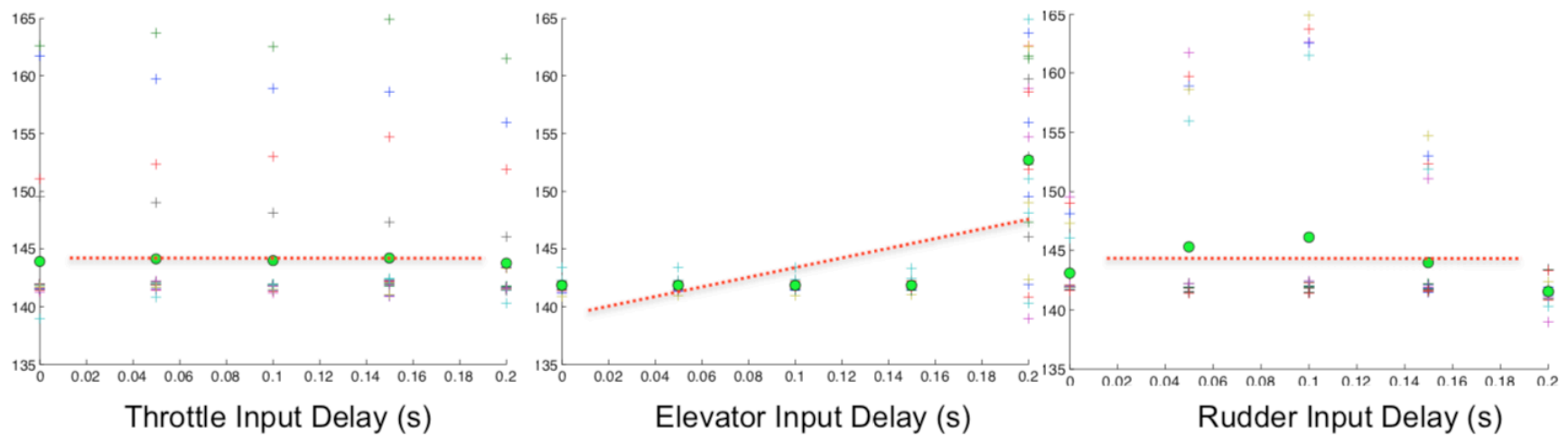


Figure 35: Sensitivity comparison plots for digital update rate attack effect on bank angle stability. The y-axis represents the square of deviation of bank angle from trim condition (deg^2). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear t line of the mean values.

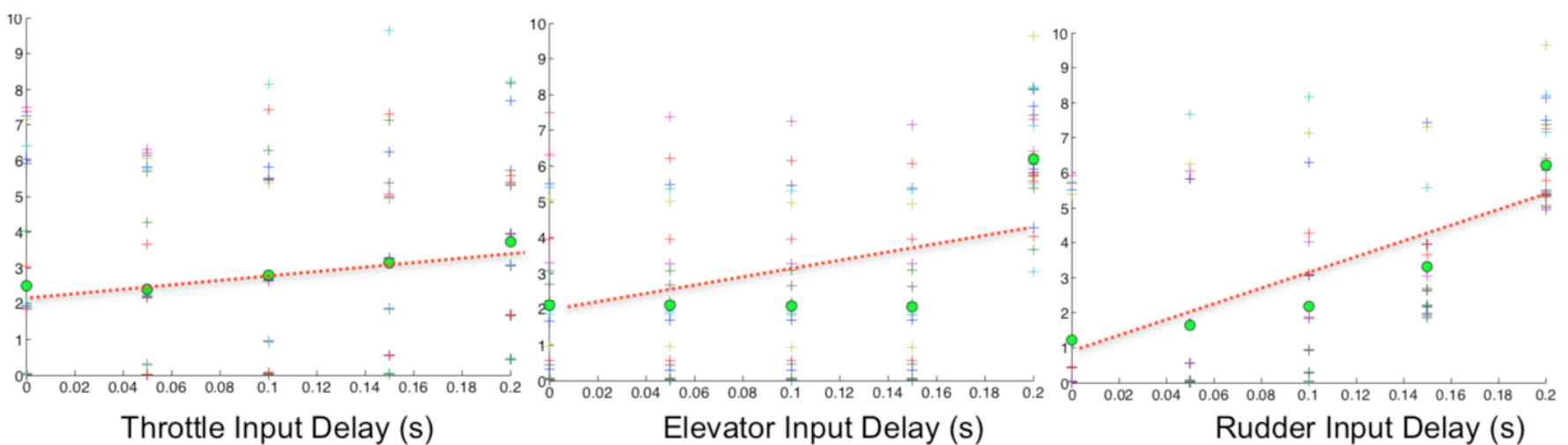


Figure 36: Sensitivity comparison plots for digital update rate attack effect on pitch angle stability. The y-axis represents the square of deviation of pitch angle from trim condition (deg^2). The green circle represents the mean of the + marks, which represent the result of each execution. The dotted line represents the approximate linear t line of the mean values.

VIII. Cyber-Secure Architecture of Autopilot System

We propose to develop a system architecture that is more robust in order to thwart these cyber attacks by augmenting the UAV autopilot system with a cyber-security monitoring component, called a Supervisor. This proposed architecture is shown in Figure 37. The Supervisor's role is to detect and isolate abnormal or malicious activity within the autopilot system. It can report over the communication link but will remain inaccessible from the outside to help ensure it maintains security. It will read and monitor the other systems without being directly in the operation loop. This will protect it from attacks and problems that may propagate through the system. It will also have the ability to recognize compromised areas and reconfigure the autopilot system based on this knowledge. [14]

If malicious code were loaded that could affect operation of the autopilot system, this would trigger the supervisor since the behavior of the vehicle would not be as expected. For example, if a piece of code is loaded that flips the direction of the aileron feedback, the resulting behavior of the vehicle would not match that of the dynamic model, and would be caught by the supervisor. The supervisor may disable this aileron and fly with just one aileron. The performance may be reduced but the vehicle is able to safely operate.

If false information were supplied, such as a false ADS-B signal, the supervisor would recognize that the signal strength does not match the signal strength model for that broadcast range. Then, using probabilistic methods, the supervisor can estimate the likelihood that the signal is incorrect and report it to the ground station or other members of the network.

Possibly the hardest attack to identify is the direct control of the vehicle through commands, changes in flight plan, or mission objective. While additional layers of encryption and authentication should be added to hinder this type of attack, it is still necessary to use the supervisor to aid in security. One idea is to monitor if these commands place the vehicle outside of an operational envelope or mission envelope based on the system model and mission model. [15] The operational envelope is the combination of limits within which the system can safely operate such as airspeed, g-force, altitude, etc. The mission envelope is the allowable deviation from the mission plan that still allows the objective to be effectively completed. If the command actually changes the mission envelope, then this could perhaps trigger a verification request by the supervisor via another secure communication method, and at the very least notify that a change has been issued.

So far we have proposed the detection of cyber attacks. We further propose to handle the reconfiguration and control of a compromised system. This control decision would be based on the mode that the system is placed into as a result of the attack. The outcome of the attack should also be considered as well. When deciding a course of action, possible outcomes could include but are not limited to:

1. Objective is still achievable
2. Objective is achievable with reduced performance

3. Vehicle is placed outside of its safe operating envelope
4. Vehicle is placed outside of the mission envelope
5. System is captured or destroyed
6. Objective is maliciously altered

An attack that still allows the objective to be reached may require no reconfiguration at all, while other outcomes may require the system to return home or in extreme cases safely remove itself from operation.

There are many ways to prevent and handle cyber attacks on unmanned systems. We propose that adding the Supervisor as an additional layer of security at the system design, architecture, and estimation and control levels will help contribute to a more robust multiple-layer security design. [16]

For completeness, Figure 37 also shows a detailed diagram of UAV autopilot processes and interactions of a micro-controller based system. This diagram identifies potential areas that could be vulnerable to cyber attacks in an architecture that includes a supervisor. It is color coded by threat level. The supervisor, guidance, navigation, and control processes all exist in read only memory. This memory is only writable when the processor is reset. A write to read only memory (ROM) can typically be triggered by a button on-board, a power cycle, or an external communication pin. Using the external communication pin would allow remotely programming the system but this is strongly discouraged as it would also allow a cyber attacker to reprogram the system. Assuming that the internal processes are not compromised, we can rely on the supervisor process to monitor reading to and writing from the Random Access Memory (RAM). This is where system critical information such as the vehicle state, flight plan, and communication buffers are stored.

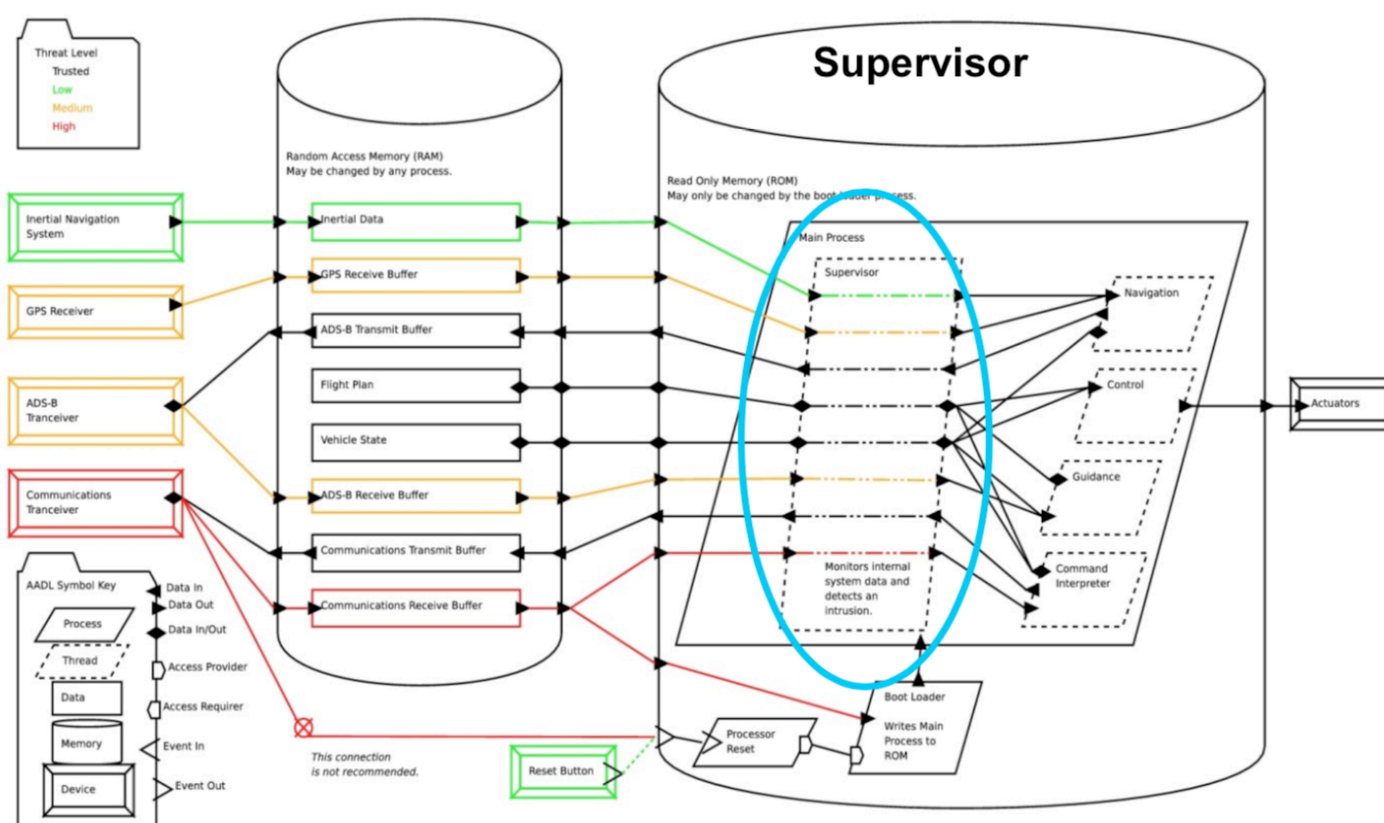


Figure 37: Data flow diagram of the UAV autopilot system with the Supervisor

IX. Conclusion

Our final goal is to develop the Supervisor to be introduced into the current UAV autopilot system, which will make the system robust enough to thwart cyber attacks. In order to do so, we have several tasks planned, which include:

Carry out GPS/INS Analysis: So far we have carried out numerical analysis on GPS attack scenarios of a simple case without coupling of aircraft dynamics. We will develop a more sophisticated and accurate model to simulate the GPS attack being applied to realistic scenarios by introducing actual aircraft dynamics. A sensitivity study for this type of attacks will also be carried out.

Carry out ADS-B Analysis: In order to perform a numerical analysis on an ADS-B attack scenario, we will first develop a collision avoidance algorithm. Then, a numerical analysis will be carried out simulating multiple aircraft. A sensitivity study for this type of attack will also be carried out.

Study more sophisticated attack scenarios: So far we have identified and analyzed simple cyber attacks that involve a single point of attack or a single method. We will study more sophisticated attacks such as the ones that utilize multiple points of attack or multiple methods. We will further look into coordinated attack possibilities where the attacker uses several attacks in a certain manner to induce more effective faults into the autopilot system. We will also consider the disguised attack possibilities where the attacker can mask an attack to induce a false reaction from the autopilot in order to remedy the attack.

Analytical analysis: We will also analytically look for cyber attack methods. For example, certain Kalman filtering algorithms might be vulnerable to a special form of induced error in measurements that cannot be detected.

Develop metrics for cyber attacks: So far a metric for measuring either a likelihood (or probability) or a damage potential of cyber attacks on a UAV autopilot does not exist. Developing these metrics is very important since it can be used to design the cyber-secure autopilot architecture.

Develop cyber attack detection algorithms: One of the roles of the Supervisor is to detect and isolate any cyber attacks. The Hybrid System's fault detection approach can be used to develop a robust detection capability.

Develop the Supervisor: With all the necessary research completed, we can move onto actually developing the Supervisor, which will play the main role in the cyber-secure UAV autopilot.

References

1. Frost and Sullivan, \Study Analysing The Current Activities in The Field of UAV," Tech. rep., European Commission Enterprise and Industry Directorate-General, 2007.
2. Yochim, J. A., The Vulnerabilities of Unmanned Aircraft System Common Data Links to Electronic Attack, Master's thesis, U.S. Army Command and General Staff College, Fort Leavenworth, Kansas, Jan 2010.
3. Army UAS CoE Sta , \U.S. Army Roadmap for UAS 2010-2035," Tech. rep., U.S. Army UAS Center of Excellence (ATZQ-CDI-C), 2010.
4. Clapper, J. R., Young Jr., J. J., Cartwright, J. E., and Grimes, J. G., \Unmanned Systems Roadmap 2007-2032," Tech. rep., Memorandum for secretaries of the military departments, Dec 2007.
5. Donley, M. B. and Schwartz, N. A., \United States Air Force Unmanned Aircraft Systems Flight Plan 2009-2047," Tech. rep., United States Air Force, May 2009.
6. Cowan, C., Wagle, F., Pu, C., Beattie, S., and Walpole, J., \Buffer overflows: attacks and defenses for the vulnerability of the decade," DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings, Vol. 2, 2000, pp. 119{129.
7. Shea, D. A., \Critical Infrastructure: Control Systems and the Terrorist Threat," Tech. rep., CRS Report for Congress, 2004.
8. Alberts, D. S., Garska, J. J., and Stein, F. P., \Network Centric Warfare: Developing and Leveraging Information Superiority," Tech. rep., DoD C4ISR Cooperative Research Program, Feb 2000.
9. Sakamoto, N. S., \UAV Development and History at Northrop Grumman Corporation Ryan Aeronautical Center," SI4000 SUMMER 2004 UAV Brief, 2004.
10. Hwang, I., Kim, S., Kim, Y., and Seah, C., \A Survey of Fault Detection, Isolation, and Reconfiguration Methods," IEEE Transactions on Control Systems Technology, Vol. 18, May 2010, pp. 636{653.
11. Krozel, J. and Andrisani, I., \Bindependent ADS-B verification and validation," AIAA 5th Aviation, 2005, pp. 1{11.
12. Godefroid, P., Levin, M., and Molnar, D., \Automated Whitebox Fuzz Testing," 15th Annual Network and Distributed System Security Symposium (NDSS), Feb 2008.

References

13. Stevens, B. and Lewis, F., Aircraft Control and Simulation, Wiley, New York, NY, 1992.
14. Liu, W. and Hwang, I., "Robust Estimation Algorithm for A Class of Hybrid Systems with Unknown Continuous Fault Inputs," American Control Conference (ACC), 2010.
15. Hwang, I., Stipanovic, D. M., and Tomlin, C. J., "Polytopic approximations of reachable sets applied to linear dynamic games and to a class of nonlinear systems," Advances in Control, Communication Networks, and Transportation Systems Systems and Control: Foundations and Applications, 2005.
16. Dufrene Jr., W., "Mobile military security with concentration on unmanned aerial vehicles," Digital Avionics Systems Conference, 2005. DASC 2005. The 24th, Vol. 2, October 2005, p. 8 pp.



Counter UAV strategies – A current day review

Alan Roder

A DJI Phantom 3 drone is shown in flight against a sunset sky. The drone is black with green accents and is positioned in the upper left quadrant of the frame. Below it, a city skyline is visible, with several tall buildings silhouetted against the orange and yellow light of the setting sun. The overall scene is a composite image used as a background for the author's bio.

ABOUT THE AUTHOR

ALAN Roder

I have been a West Midlands Police Officer since 2008 and have worked on investigation teams and as a Digital Media Investigator; I currently work within the Digital Forensics unit as a Forensic Officer analysing digital devices such as computers, storage mediums and most recently UAVs.

I graduated from the University College Dublin with a Master of Science degree in Computer Forensics and Cybercrime Investigation. I co-authored my first paper in 2018 entitled 'Unmanned Aerial Vehicle Forensic Investigation Process: DJI Phantom 3 Drone as a Case Study'. I am currently continuing my research and development in regards to UAV Forensics and hope to write further academic papers on this topic.

Unmanned Aerial Vehicles are, without a doubt, an emerging market, and their growth as a sector has the potential to revolutionise the world. Growth in this area is apparent in the private sector, public sector and military sector, with financial investment in innovation reaching levels similar to that generated by small countries.

World renowned organisations, such as PricewaterhouseCooper (PWC) [1], have recognised the importance of UAVs to industry and published reports identifying business considerations and risks. Additionally, academics and researchers are continually completing academic research and producing papers covering forensic techniques, guidelines and evidential parameters [2,3,4,5,6,7,8].

However, considering all of this, there is still one element which is struggling to draw level with demand, and that is automated counter UAV (CUAV) strategies and monitoring solutions.

Challenges in identifying and monitoring UAVs

One of the greatest challenges associated with UAVs, is the ability to accurately monitor and track their movements. UAVs, in particular Small Unmanned Aerial Systems, are no bigger than an average bird and can be as small as an insect. As a result of this, using traditional radar systems would likely result in identifying too many false positives.

Additionally, Manned Monitoring Systems can be costly to operate, with the requirement for 24 hour cover, shift patterns, holiday pay and other considerations when employing staff. Given the increase in UAV activity, the only viable option is to create, design and develop an automated technology.

The current market has four main types of UAV monitoring equipment:

Radar

Radar uses radio energy to detect objects. It sends out multiple signals and when it receives a reflected signal, it measures the direction and distance from the target. Most radars send their radio signals as a burst, and effectively listen for the sound coming back, similar to an echo.

Traditionally, radars are specifically designed to disregard small targets, as the results could contain a high percentage of false positives, but they are perfectly designed to track large objects like aircrafts etc.

- **Positives:** Radar is designed for long range and constant tracking, with a high level of accuracy. They can handle a large number of targets simultaneously, and can track all UAVs irrespective of autonomous flight. Radar is not affected by visual conditions, such as the weather or daylight.
- **Negatives:** The detection range is dependent on the size of the UAV, and most systems cannot distinguish birds from UAVs. Additionally, there may be frequency disruption interference.

Radio Frequency Analysers

Radio Frequency Analysers consist of one or more antennas which receive radio waves, along with a processor to analyse the RF spectrum. These systems are used to detect radio communication between a UAV and its ground control station.

Most systems are able to identify the most common UAV makes and models, for example DJI, but there are some which are capable of further identifying the MAC addresses of the UAV and Ground Control Station, assuming the UAV uses Wi-Fi communication.

The MAC address, or Media Access Control address, is given to a network adapter when it is manufactured. It is hardwired or hard-coded onto the UAVs Network Interface Card and is unique to it.

Identifying the MAC address is considered extremely useful when attempting to prove that a UAV and Ground Control Station were paired and active. Some high-end systems are also able to triangulate the UAV and Ground Control Station when using multiple radio units spread far apart.

- **Positives:** RF analysers can often have a low monetary cost. They have the capacity to detect multiple UAVs and Ground Control Stations, and are usually a passive detection system. Some are able to triangulate the UAV and Ground Control Station coordinates.
- **Negatives:** RF analysers cannot track all UAVs. Additionally, an RF frequency is necessary, so they cannot detect autonomous UAVs. They also struggle to differentiate frequencies in crowded RF areas, and typically have a short range.

Optical Sensors

Optical sensors are in essence, a video camera or multiple video cameras working in synchronisation. Systems include standard cameras, but more efficient systems usually include infrared, thermal and night vision imaging.

- **Positives:** Optical sensors provide visual detection of a UAV and potentially its payload, and can record images which can later be used as forensic evidence.
- **Negatives:** Optical sensors can be difficult to use, purely for detection, and can provide a high number of false positives. They also have poor performance in poor weather conditions and dimly lit or dark environments.

Acoustic Sensors

Acoustic sensors typically consist of a microphone, or microphone array, which can detect the sound made by a UAV and calculate a direction. It is possible, with multiple arrays, that acoustic sensors can be used to triangulate a rough position of a UAV.

- **Positives:** Acoustic sensors are typically a medium cost investment, which provide directional information, but not normally a specific location.
- **Negatives:** Acoustic sensors do not normally work well in noisy environments, and they are short range.

There are several UAV identification and monitoring systems which are currently on the market, however all of these have varying levels of accuracy.

One of the most popular currently, is the AeroScope [9], which states to be a drone detection platform. The company have worked closely with DJI and indicated that the system may be able to take control of a UAV, however this notion has since been removed from the website.

DJI are currently the market leading supplier of commercial UAVs worldwide. As a result of this Aeroscope have a significant advantage over other global competitors, however the accuracy of their software still needs to be considered in particular in regards to their detection of other UAV manufactures platforms, for example Yuneec and Parrot.

Another company which have recently entered the market are CRFS [10]. CRFS are not a new company and primarily focus on designing, building, programing and deploying systems and solutions for RF spectrum monitoring, management and geolocation.

The system is based around a network of RFeye nodes that are set up around a target facility. These nodes passively detect and identify the presence of RF transmissions that relate to UAVs, even under sub-optimum conditions. The transmissions can be geo-located in 3D to give the location of the UAV, including flight height and air speed. Multiple UAVs can be simultaneously tracked and identified.

Dronesshield [11] have taken the next step in UAV detection and response, and provided an all in one solution. It provides a software engine with proprietary multi-sensor detection technologies which detect small UAVs in three dimensions.

Once identified, the system has countermeasures which allow for the controlled management of UAV payloads such as explosives, with no damage to common UAV models or the surrounding environment. Countermeasures have been developed for long range coverage in a wide range of environmental conditions.

Ultimately there is not detection system currently on the market which accurately and consistently tracks all UAVs.

Counter UAV strategies

It would be true to say that UAV counter measures are an evolving industry, and very much in its infancy. Companies are constantly pushing to create safe and reliable methods to counter rogue UAVs, with most major governments offering financial contracts to anyone who presents a reasonable business proposal.

This provides both opportunity and threat, as it is clear that no one technology is able to effectively counter all forms of Small Unmanned Aerial Vehicle.

When considering UAV countermeasures, there are three main overarching categories;

- **Physically destroying the UAV** – The simplest method which has the highest risk of collateral damage and the lowest chance of identifying the pilot.
- **Neutralising the UAV** – Similar to destroying the UAV, but usually incorporates a retrieval or controlled landing of the rogue UAV.
- **Taking control of the UAV** – The hardest solution, however provides the least risk of collateral damage, and the highest potential for identifying the pilot.

The markets eight main CUAV platforms:

Birds of prey

Whilst very much a low tech solution, eagles have been trained to capture UAVs. Eagles are natural hunters, and this solution takes advantage of the natural instincts of the species. This solution does however, require a lot of investment for the training and maintaining of the birds.

- **Positives:** If the birds of prey are available in your area, interception a UAV can be quick and accurate with low risk of collateral damage.
- **Negatives:** It can be difficult to scale up or down as there are few birds of prey trained in this specialism. Additionally, this resource requires significant investment into training and maintenance. If used near airports, the birds themselves could prove to be a hazard.

Nets & net guns

A net would stop a UAV by restricting the blades and preventing them from moving. There are three main types of usage:

- **Net Cannon fired from the ground:** These can be hand-held, shoulder mounted and or a fired turret. This system is effective anywhere from 1m to 150m, however accuracy would diminish over distance. Nets can be fitted with or without a parachute for controlled descent of the captured UAV.
- **Net cannon fired from another UAV:** This system can overcome what can be a limited range compared to a net cannon fired from the ground, however it can be difficult to capture another moving target. Due to the

likely height this system would work at, they are normally fitted with a parachute for controlled descent of the captured UAV.

- **A net hanging from a deployed UAV, otherwise referred to as a ‘net drone’:** The target UAV is captured by manoeuvring the net towards the rogue UAV. The ‘net drone’ would normally be capable of either carrying the rogue UAV to a safe area, or release the captured UAV with or without a parachute.
 - **Positives:** Nets can capture UAVs, allowing for good potential for forensic recovery. Ground launched net cannons can be semi-automatic and with a high degree of accuracy. Additionally, UAV deployed nets have a long range, with a low risk of collateral damage.
 - **Negatives:** If there is not a controlled descent of the target UAV, then this could result in debris and collateral damage. UAV deployed nets can be imprecise and have a long reload time. Finally, ground launched nets have a relatively short range.

RF Jammer

An RF Jammer can be static, mobile, or a handheld device which transmits a large amount of RF energy towards the UAV, which confuses the UAV into losing the Ground Control Stations signal. The result of which depends on the UAV and results in either:

- The UAV defaults into making a controlled landing in its current position
- The UAV defaults into returning to the pilot or the home location.
- The UAV falls uncontrolled to the ground.
- The UAV flies in a random and uncontrolled direction.
 - **Positives:** RF Jamming systems are a medium cost solution, providing neutralisation without damaging the target UAV.
 - **Negatives:** RF Jamming is short range, and may affect other radio communications. Depending on the make and model of the target UAV, the result can prove to be unpredictable and could unintentionally cause more damage than would otherwise have occurred.

GPS spoofing

GPS spoofing is the act of sending a new signal to the UAV, replacing the communication with GPS satellites it would use for navigation. This method effectively ‘spoofs’ the UAV into thinking it’s somewhere else.

By altering the GPS coordinates in real-time, the UAVs position can be controlled by the spoofer. Once control is gained the UAV can be directed to a safe location for retrieval.

- **Positives:** GPS spoofing is a medium cost solution, providing neutralisation without damaging the target UAV.
- **Negatives:** GPS spoofing is short range, and may affect other radio communications.

High Power Microwave

High Power Microwave devices generate an Electromagnetic Pulse or EMP, which is capable of disrupting electronic devices. The EMP interferes with radio links and can disrupt or even destroy the circuitry in UAVs due to the damaging voltage it creates. High Power Microwave devices can include a focusing antenna so that the EMP can be targeted in a certain direction, thereby reducing potential collateral damage.

An EMP pulse has the potential to cause significant disruption to electrical systems, and whilst most critical infrastructure will have integrated measures to defend against such an attack, many important organisations, such as hospitals, will not have protection.

- **Positives:** Should the UAV be within range, it can be stopped efficiently.
- **Negatives:** EMP generators are expensive, additionally, EMP pulses are usually omnidirectional, which means there is a significant risk of unintentionally disrupting communications or destroying other electronic devices in the area. And finally, any affected UAV would likely fall in an uncontrolled descent, which could cause collateral damage.

High-energy lasers

High energy lasers, are high powered optical devices which produce an extremely focused beam of light, or laser beam. The laser defeats the UAV by destroying the structure or the electronics.

- **Positives:** High energy lasers are an effective method of physically stopping a UAV.
- **Negatives:** High energy lasers are expensive, and are mostly an experimental technology. Additionally, any affected UAV would likely fall in an uncontrolled descent, which could cause collateral damage.

Ballistic UAVs

UAVs designed to intercept other UAVs mid-flight and collide with them. Interception can be either manned or automated.

Ballistic UAVs have the potential to intercept any threat. They have the potential to be fitted with small explosive charges, thereby effecting an accurate strike.

- **Positives:** Effective against all types of UAV. Relatively low cost when considering the potential damage or loss of life. Highly accurate. Scalable when involving multiple UAVs.
- **Negatives:** Flight time to target may be too slow. Additionally, any effected UAV would likely fall in an uncontrolled decent, which could course collateral damage.

Rifles and other Firearms

Automated or semi-automated rifles designed to shot a UAV out of the sky.

- **Positives:** Often highly accurate. Minimal targeting and travel time.
- **Negatives:** High risk of collateral damage, both from the UAV and the round which would continue through the UAV.

Some of these defensive systems may seem farfetched and not practical, however, an article by Defence News [12], published in July 2019, stated that the UK Ministry of Defence are planning to invest up to £123 million developing three directed-energy weapon demonstrators, including one aimed at killing UAVs. The demonstrators are part of the MoD's *'Novel Weapons Programme,'* which is responsible for the trial and implementation of innovative weapon systems. The new arms are expected to reach the frontline within 10 years.

When considering any counter UAV strategy, attention must be given to collateral damage, especially when defending locations situated in built up areas. Many of the CUAV strategies mentioned above, are designed to destroy the UAV mid-flight, and whilst they may be effective, could have lethal consequences to those who may have a UAV fall on them.

Additionally, many of the targeted CUAV solutions, such as High-Energy lasers, need to consider that a projectile or laser will not necessarily stop once the target has been struck. A round from a high powered rifle could travel over a mile further than the intended target, which may strike a passing aircraft.

UAV crime prevention strategy

Countries and nation states are coming to realise the threats associated with unrestricted UAV flight, especially when taking into account airspace around airfields, military installations, national infrastructure and critical locations, such as power stations.

Whilst the technologies and solutions discussed above will be necessary, ultimately it would be impossible to maintain the current freedoms of relatively unrestricted flight, whilst also maintaining security, without a framework for lawful use.

However it is important to note that the rights of those wishing to continue their hobby, sport and business should be protected, without unnecessary cost or curtailment of freedom.

Given all available research on this topic, the only viable option is to create a multi-level legal framework, designed to maintain freedom of lawful flight, whilst also protecting sensitive locations and flight paths.

UAV Pilot's Licence:

Whilst this concept may seem unachievable in the current climate, it is obvious that the regulation of the use of UAVs will ultimately require implementation i.e. through the use of licencing at some point in the future. This will promote the safe and lawful use of UAVS; whilst enabling a framework for executing penalties for illegal or dangerous behaviour.

When cars were first produced in the United Kingdom in 1892 there was no requirement for a licence, which was only introduced in 1903. Incidentally, driving tests were not started until 1935.

The concept of introducing a licence was not popular, however the UK government anticipated their use and decided to introduce the licence in an effort to identify vehicle use. The driving test was later introduced to reduce road fatalities.

One must assume that the use of UAVs, either through manual flight or automated flight paths, will become a standard sight in our skies, with companies such as Amazon researching automated package delivery through this medium [13].

UAV operators have the potential to become a recognised career path, much in the same way as taxi drivers are today. A method of identifying those who are trained and insured to carry out these tasks will become a necessity to maintain control and promote safe flight.

By creating the legal requirement for all UAV pilots to hold a nationally recognised licence, will promote safe practice, whilst also supporting law enforcement agencies in identifying and prosecuting illegal activity.

UAV insurance and registration:

Car insurance became mandatory in the United Kingdom under the Road Traffic Act in 1930, which ensured all vehicle owners had to be insured for their liability for injury and/or death. Its enactment was brought about to reduce road fatalities through safer practices, whilst also linking an individual to a vehicle for the purpose of recompense to an injured party.

This legal requirement had the additional benefit of associating an owner to the vehicle, thereby putting the responsibility of the owner to account for the vehicles whereabouts at any given time. As such, it became less likely that the owner would commit crime in their own vehicle, as it would be registered to them.

By creating mandatory UAV insurance the same principle would apply and owners would be less likely to use UAVs registered to them to commit illegal or dangerous activities.

Car registration plates were first introduced in the United Kingdom in 1904 under the 1903 Motor Car Act. When a vehicle was registered the owners name and address were recorded and matched with their driver's licence.

Whilst easily circumvented at that time, it created the framework for vehicle registration used today and is used to ensure vehicle tax is paid and Ministry of Transport (MOT) tests are maintained.

By making it a mandatory process that all UAVs are registered, the owner would become liable for its use. Law enforcement would then have additional powers to seize UAVs where there is no registered owner or insurance.

A paper by Mais Nijim and Nikhil Mantrawadi entitled '*Drone classification and identification system by phenome analysis using data mining techniques*' [14] supports the assertion that UAV classification and identification is necessary and provides avenues for creating such a system.

Any positive introduction to regulate lawful use of UAV's will inevitably breed negative connotations, such as the potential theft of UAVs for the participation in criminal activity. Since this practice of theft is likely in occurrence, registration and insurance would only help to support victims of crime.

By creating a UAV pilots licence, with the supporting framework of insurance and registration, the opportunity to commit crime with relative anonymity would be greatly reduced.

Subscriber Identification Module (SIM) and Media Access Control Address (MAC)

The simplest method of monitoring the lawful use of UAVs is to enforce a requirement that all UAVs be fitted with a SIM card and that the mainboard be registered with a MAC address.

By requiring that all UAVs contain a SIM card and MAC address, UAVs can be easily monitored, with law enforcement only really needing to focus on the rogue UAVs and/or UAV devices suspected of being used for illegitimate purposes.

Many argued that the government would be spying on road uses when the United Kingdom introduced the ANPR system (Automated Number Plate Recognition). The system captures the number plate of a vehicle passing by, which can be accessed by law enforcement agencies when investigating criminal offences. The system is now largely ignored by the public, who accept it as a necessary technology.

Given that UK airspace is one of the busiest in the world, it is unlikely any one pilot would be tracked. However, being able to accurately track UAV flight paths would indicate malicious activity and potentially provide valuable information to assist in a lawful investigation.

This system does not have to be expensive, with contract and ‘Pay as you Go’ data plans selling for less than £5 per month on the market currently. Given the cost of UAV platforms, the additional data cost would be marginal.

Opportunities and limitations:

It cannot be denied, that Policing and Military budgets globally are under strain in the current economic climate, as such any reference to the reduction of target vulnerabilities has to be given with due consideration.

The simplest and most cost effective solution, is for governments to authorise research into country wide detection systems which can be implemented by all law enforcement forces and military bases. The system would also have an oversight in the interest of national infrastructure.

This research into UAV detection technology would likely work with civilian agencies for example airports, who have already created localised detection systems. This would reduce the initial cost, although likely increase the long term expenditure as organisations look to update or streamline their technology.

This system, once in place, would work alongside a UAV registration database with the aim of identifying UAVs which have not been registered, whilst also identifying registered UAV flights and tracking their activity. Any breach in lawful activity would be notified to local law enforcement to conduct enquiries and act as appropriate.

The system would be used by the military to identify breaches of restricted airspace, with sufficient warning for a proportionate response to be generated against the threat.

By increasing the monitoring of airspace and bolstering the defence of key military installations and locations or national infrastructure, the risk of an attack can be reduced.

Conclusion

UAVs have proven to be an excellent resource when used as a proactive tool by emergency services and other law abiding agencies and individuals.

The increased use of UAVs is almost inevitable, fuelled by the desire to move into a future stereotyped in films such as ‘Ready Player One’ and ‘Ghost in the Shell’ and games like ‘Half Life 2’ and ‘Cyberpunk 2077’.

Rather than try to prevent their use, it is apparent that it would be extremely productive to work with industry experts and enthusiasts to design a framework for lawful use which is robust and achievable for law enforcement agencies to

act on. Any action by government should encourage proactive and lawful use of UAVs, which would potentially promote a level of ‘self-policing’ within the community.

A research project conducted by the University of Birmingham entitled ‘*Nefarious Criminal and Terrorist Uses of Unmanned Aerial Vehicles (UAVs) (Apr 2016 - Apr 2017)*’ has concluded that:

- There are insufficient counter-measures in place.
- Existing laws are not being enforced.
- The HMG (Her Majesties Government) are reluctant to introduce new legislation.

A globally unified approach for intelligence gathering and sharing is essential for each nation state, and necessary for accurate and intelligence lead dissemination of information. In reality, few agencies have the capacity to achieve this aim, due in part to cost, but also due to the decreased cooperation seen in recent years, with the exception of Interpol who encompass 192 countries.

It is hoped that governments continue the collaboration with their foreign counterparts in the sharing and dissemination of information. For example, global law enforcement agencies such as Interpol and Europol can commission a research project into creating an UAV intelligence database, which will assist in identifying crime trends, strategic weaknesses and support the identification and prosecution of offenders.

With regard to UAV countermeasures, it is important that companies understand the market that is most likely to purchase their systems and infrastructure. For example, likely purchasers will include government agencies, military services and law enforcement, along with those wishing to protect national infrastructure including airports and power stations.

Whilst the primary objective will always be to prevent harm to persons and damage to property, the secondary objective will inevitably be to identify the offender and gather evidence to prosecute where possible.

As such countermeasures should aim to gather data in addition to control and/or neutralisation of the threat.

Gathering the UAV make and model, along with any serial numbers would help law enforcement link any UAV found during a search, to devices linked to crimes. Additionally, extracting any stored flight data and media would help identify and locate suspects. In the medium to long term a national database could be created which would store this information, which could then be accessed by law enforcement agencies. Should a device be seized, the serial number could then be searched on the database to see if it had been identified in a previous crime.

All of the information and ideas explored in this article forms a small portion of the papers I have written and the course I designed, using first hand law enforcement experience, along with academic research. It is my hope that you feel encouraged to take the course: <https://eforensicsmag.com/product/drone-forensics-w44/>

References

1. PWC. (N/K). *drones*. Available: <https://www.pwc.co.uk/issues/intelligent-digital/drones.html>. Last accessed 04/03/2020.
2. Alan Roder et al. (2019). *Unmanned Aerial Vehicles (UAVs) Threat Analysis and a Routine Activity Theory Based Mitigation Approach*. Available: https://link.springer.com/chapter/10.1007/978-3-030-31239-8_9. Last accessed 04/03/2020.
3. Alan Roder, Raymond Choo and An Lekhac. (2018). Unmanned Aerial Vehicle Forensic Investigation Process: DJI Phantom 3 drone as a case study. Available: <https://arxiv.org/ftp/arxiv/papers/1804/1804.08649.pdf>. Last accessed 14/12/2018.
4. David Kovar. (2018). *Defending Against UAVs Operated by Non-State Actors*. Available: <https://integriography.files.wordpress.com/2017/03/david-kovar-gmap-16-thesis.pdf>. Last accessed 04/03/2020.
5. Birmingham University. (2014). The Security impact of drones: Challenges and opportunities for the UK. Available: <https://www.birmingham.ac.uk/Documents/research/policycommission/remote-warfare/final-report-october-2014.pdf>. Last accessed 06/02/2019.
6. Richard Duffy. (2018). Understanding the public perception of drones. Available: <https://www.nesta.org.uk/blog/public-perception-of-drones/>. Last accessed 07/02/2019.
7. Cohen LE, Felson M. Social change and crime rate trends: a routine activity approach. *American Sociological Review* 1979;44(4):588e608.
8. Bill Goodwin. (2018). Drones: How the world's airspace will change in 2019. Available: <https://www.aircargonews.net/news/technology/single-view/news/drones-how-the-worlds-airspace-will-change-in-2019.html>. Last accessed 24/01/2019.
9. Aeroscope. (N/K). *Aeroscope*. Available: <https://www.dji.com/uk/aeroscope>. Last accessed 04/03/2020.
10. crfs. (N/K). *drone-detection*. Available: <https://www.crf.com/drone-detection/#>. Last accessed 04/03/2020.
11. droneshield. (N/K). *droneshield*. Available: <https://www.droneshield.com/how-droneshield-works1>. Last accessed 04/03/2020.
12. Andrew Chuter. (09/07/2019). *uk-shoots-for-new-laser-weapons-against-drones-missiles*. Available: <https://www.defensenews.com/global/europe/2019/07/09/uk-shoots-for-new-laser-weapons-against-drones-missiles/>. Last accessed 04/03/2020.
13. BBC. (2016). Amazon makes first drone delivery. Available: <https://www.bbc.co.uk/news/technology-38320067>. Last accessed 07/02/2019.
14. Mais Nijim, Nikhil Mantrawadi . (2016). Drone classification and identification system by phenome analysis using data mining techniques. Available: <https://ieeexplore.ieee.org/abstract/document/7568949>. Last accessed 06/02/2019.



Protecting the Unmanned Aerial Vehicle from Cyberattacks

Jesus Nunez

Vincent Tran

Ajay Katangur

ABOUT THE AUTHOR

JESUS NUNEZ

Jesus Nunez is a fifth-year computer science student at California State University Dominguez Hills, was mentored by Dr. Ajay Katangur.



ABOUT THE AUTHOR

VINCENT TRAN

Teaching Assistant at CSU Dominguez Hill



ABOUT THE AUTHOR

AJAY KATANGUR

Georgia State graduate Ajay Katangur has been named head of the Computer Science Department at Missouri State University. Dr. Katangur is currently an associate professor in the Department of Computing Sciences at Texas A&M University-Corpus Christi. He will start his new position on August 1. Missouri State is a public university located in Springfield, Missouri. With an enrollment of 26,000, it is second-largest university in the state.

The increased use in drone technology has made them a popular option for companies to use when performing certain tasks. Due to this increase in popularity, security analysis has become crucial. In this article, attacks performed and observations associated with security vulnerabilities in the AR Parrot 2.0, 3DR Solo, and the DJI Phantom 4 Pro drones will be presented. The current auto-pilot systems and security protocols will also be examined for vulnerabilities and cyberattacks that are common in network systems. Currently the AR Parrot 2.0 drone communicates through an open Wi-Fi connection, making it vulnerable to multiple forms of attacks. The 3DR Solo works through a password protected Wi-Fi signal, however, it is possible to obtain such password with the use of specific tools discussed later in the paper. This poses a potential threat to the system, leaving it open to intrusion. Although the DJI has improved security compared to previous models, GPS spoofing still remains a viable form of attack making it a possible vulnerability in the system. Other forms of attacks will also be explored to see if there is more beyond tampering with the network communications of the drones that might pose a threat to the system.

I. INTRODUCTION

A. Motivation

UAVs are typically controlled from the ground by remote control that use radio waves for communication. For a hacker, it is not difficult to jam the radio waves, thereby effectively making the person lose control of the UAV. Each of the drones examined in this paper function differently so the approach taken in looking for vulnerabilities will vary. As long as the Wi-Fi signal the drone communicates in is able to be picked up by the hacker, the drone is vulnerable to an attack. Tools already exist that facilitate attacks on drones. Icarus [4], a tool used to grant an attacker complete control over a target drone, allows the attacker to steer, accelerate, brake and even crash them. SkyJack, an open source drone-hijacking program, can also be used to seek out and wirelessly take over other drones within range, essentially allowing the attacker to create an army of zombie drones all under their control. For these reasons, it is critical to analyze and address the security of aerial vehicles especially those controlled autonomously through a wireless connection.

Unmanned Aerial Vehicle (also known as a drone) technology is rapidly growing in popularity [1]. Primarily used by the defense department, UAVs are now utilized for non-military tasks putting them at high demand by both average consumers or hobbyists, and commercial businesses like Google and Amazon. Companies are already utilizing drones for specific tasks such as delivery services, agriculture, infrastructure development, aerial surveying, and tasks that might be dangerous for a human to perform [2]. As a result, UAV traffic is going to increase in the coming decade, and with it, the possibilities of cyberattacks will rise. UAVs can be hijacked, their paths can be changed, and they can be made to collide with other UAVs or objects. Furthermore, if they are equipped with a weapons system, they can be maliciously utilized to fire in non-hostile situations. Commercial aircraft, although not a UAV, is still an aerial vehicle and it currently does not have a solid defense against cyberattacks. The Department of Homeland Security (DHS) states that the commercial aviation backbone is built upon a network of trust. This is not very reassuring, considering

people's lives are only protected by trust when in the air. Boeing estimates a 20 year plus service life for its current aircraft which means 15 to 20 more years of cyber vulnerability [3].

B. Security Vulnerabilities in Unmanned Aerial Systems (UAVs)

The AR Parrot 2.0, 3DR Solo, and DJI Phantom 4 Pro drones are analyzed for vulnerabilities in this paper. Attempts to exploit open connections of the AR Parrot drone, revising the application program interface (API) of the 3DR Solo to find potential weaknesses that might allow the tampering of flight plans and configurations, and exploring other options of attacks besides GPS spoofing of the DJI Phantom 4 Pro will also be tested.

Wi-Fi issues: The AR parrot drone communicates through an open Wi-Fi Protected Access (WPA) connection, making it possible for multiple users to connect simultaneously and, as a result, making it impossible to determine who is the valid original user. The 3DR Solo has more security by requiring the use of a password to access the connection between the controller and the drone. If the attacker is able to obtain the password, it is possible to have multiple users connected and, without the original user knowing, is able to change settings and deauthenticate the original user by either changing the password or using tools like aircrack-ng[5].

GPS based attack: Higher end drones use GPS signals for navigation. The GPS signals used by civilian drones are not encrypted, making them vulnerable to spoofing and other GPS based attacks. A more detailed example of the steps required to spoof the GPS and take over a UAV are presented in [6]. Currently, the DJI Phantom 4 Pro has no public access to the drones API for security reasons. This means that the only way an attacker can cause harm to the drone is through spoofing of its GPS signals while airborne.

The rest of the paper is organized as follows: Section II provides some related work and details on the influence it has had on this research. Section III offers insight on the experiments that have been performed on the specific drones. The results of these experiments are also provided in this section. Section IV is the conclusion along with future work.

II. RELATED WORK

The research on the vulnerabilities of Unmanned Aerial Systems done in Ben-Gurion University of the Negev in Israel confirmed the vulnerabilities discovered in the AR Parrot drone. Attacks pertaining to the drone's open Wi-Fi, deauthentication of the original user, exploiting the open file transfer port (FTP) to manipulate data, and snooping into the Wi-Fi and packet capturing were all performed in this article. The attacks done on the DJI Phantom 4 by this research team have provided interesting insight to theoretical attacks that might be possible. Considering, however, that most of the attacks done involve manipulation of software through direct access of the drone's hardware, they are not efficient since the supposed hacker will never have direct access to either the software or hardware of the drone. These attacks, although they suggest a potential vulnerability, might not actually be possible to be performed by an attacker. In an ideal scenario, the hacker has direct access to the drone and its software, however, the chances of that occurring are not high. For this, their research done on the DJI Phantom 4 only provides options to explore if one had

direct access to the system. The only possible attack performed that proves to be a vulnerability in the DJI Phantom 4 is the spoofing of its GPS signals using specialized tools. More specifics on this research in [7].

III. EXPERIMENTS AND RESULTS

This section will be broken down into three subsections, each describing what tests have been performed on the specific drone along with the results of each test.

A. AR Parrot 2.0 Drone

The AR Parrot 2.0 drone, shown in figure 1, is an easy to fly drone. A mobile device along with the "Freeflight Pro" application are required for control and communication of the drone. The AR Parrot 2.0 drone is a Linux-based quad-rotor that contains a HD 720p camera, and a Wi-Fi access point. The attacks performed on this drone are focused on the open Wi-Fi connection. This specific model does not offer the ability for the user to add a password to the connection to increase the security of the drone. Due to this, the overall security of the drone is minimal and its vulnerability to an attack is high.



Fig. 1. AR Parrot 2.0 Drone

1. **Open Wi-Fi:** The AR Parrot 2.0 drone uses an IEEE Wi-Fi 802.11n signal. The IP address of the drone is 192.168.1.1, and since the connection is not protected by a password, all communications are unencrypted. The drone allows for multiple users to connect at once and, with no way to validate who the original user is, anyone can control the drone. This is a massive vulnerability in the system and unless the user is given the option to protect the drone's connection with a password, the drone is prone to an attack.
2. **Deauthenticating Owner:** The AR Parrot 2.0 drone is flown using the Free Flight mobile application. Using tools such as aircrack-ng [5], it is possible to scan for current Wi-Fi networks in the vicinity and then attack a specific one using its unique MAC address. For the Parrot drone, it is not hard to break into the connection since it is not password protected. Once the handshake between the computer and the drone is

established, the deauthentication procedure takes place. This disconnects the original user, making it possible for the attacker to take over using a separate mobile device.



Fig. 2. 3DR Solo Drone

B. 3DR Solo

The 3DR Solo consists of a controller and drone components as seen in figure 2. A mobile phone can be used as part of the set up for video streaming, however, it is not required to fly the drone. Both the 3DR Solo and the controller contain an embedded Linux system. The drone specifically contains a camera and gimbal, as well as GPS. The controller has an access point that connects to the Solo and a mobile device via Wi-Fi. The connection is a password protected connection so breaching into the connection is not as easy like with the Parrot drone. The test performed required the use of aircrack-ng, and a dictionary file with possible passwords to facilitate the attack. Theoretically, the attack should not require a dictionary file, but for this test, it is utilized to test for a possible vulnerability.

1. **Deauthentication of Owner:** For the 3DR Solo, the controller is the root of all the connections and communications with the drone and mobile devices. The procedure for this attack involves first obtaining the drone's password, and then utilizing aircrack-ng to deauthenticate the original user. The result of this attack causes the drone not to crash, but to return home, the initial location from where it took off. After it lands, the connection between the drone and the controller is disrupted until the attack is stopped.
2. **Force Landing Using Mobile Device:** The 3DR Solo Drone controller works as a router for the connection communication to the drone. The connection is password protected, the default password being "sololink". The assumption is that most users will not change the password and keep the default, therefore, in most cases, the 3DR Solo's password will be the default. With this assumption, the method of attack is as follows: using another mobile device, it can be connected to the 3DR Solo network. This is an easy task due to the fact that the password is the default. Once the connection is established, two mobile devices are now connected to the one controller. From here, using the attacking device, the password can be changed and this would force the drone to land and the original mobile device to disconnect. This method allows the

attacker to force land the drone and disconnect the original user from the network. This, however, can only be performed if the initial assumption is true, the user did not change the password from the default on their device.

3. **Using Two Drones to Spoof Controller:** For this procedure, two drones and one controller are utilized. The assumption here is that by deauthenticating the original drone, using the second drone to send a pairing signal, the controller will be spoofed and connect to the other drone rather than the original drone. The second assumption is the following, once the original drone is successfully deauthenticated, the pairing signal is sent from the other drone. The end result is a prolonged connection time between the pairing of the controller and the original drone. This would allow an attacker to gain much more time for an attack should the need for it be required.
4. **Analyzing 3DR Solo APK:** By first obtaining the 3DR Solo APK (Android Package Kit) file, the .dex files can be converted to .jar, which can then be used to see the source code of the application. Once the .dex class files have been converted to .jar files, the code can be seen using a Java Decompiler tool. All tools and steps to do the following are in [8]. The Java Decompiler can only be used to see all the source code of the application, but not change it. The code that allows the linking between the drone and the controller is shown which provides a better understanding of the process undergone when the drone is trying to communicate with the controller. The purpose of analyzing the source code is to see if there might be any vulnerabilities in the way the application handles errors and exceptions regarding the connection link between the drone and the controller.



Fig. 3. DJI Phantom 4 Pro

C. DJI Phantom 4 Pro

The DJI Phantom 4 Pro, shown in figure 3, contains a controller and mobile device, along with the drone. The mobile device uses the DJI GO 4 application, which acts as the interface to all the drone's controls and settings, also providing display to the live video feed.

GPS Spoofing: DJI Phantom drones are GPS based drones. GPS allows for better navigation, and due to the fact that the drone functions under civilian GPS signals, the lack of encryption makes it easier to spoof the GPS. To spoof a GPS receiver, a transmitter is used to send false GPS signals, forcing the GPS on board the drone to synchronize with the attacker's signals. If done successfully, this enables the attacker to hijack the drone and place it at a desired location.

IV. CONCLUSION AND FUTURE WORK

In this paper, the AR Parrot 2.0, 3DR Solo, and the DJI Phantom 4 Pro drones have been analyzed for security vulnerabilities. The AR Parrot 2.0 drone is the most vulnerable of the three, making it highly prone to an attack. Unless the manufacturers allow for the user to add a password to the network the drone communicates on, this drone is too risky for any personal or commercial use. The 3DR Solo drone proved to be a far more secure drone than the AR Parrot 2.0. The ability to add a unique password to the network the drone communicates on adds a layer to the security for any user. In this paper, the lack of specific equipment limited the ability to further analyze other communication protocols in the 3DR Solo and DJI Phantom 4 Pro. In future work, radio frequency analysis will help better understand these protocols, and the use of SDR equipment, such as HackRF One [9], will facilitate this task.

REFERENCES

1. D. Joshi, "Exploring the latest drone technology for commercial, industrial and military drone uses," *Business Insider*, <http://www.businessinsider.com/drone-technology-uses-2017-7>, 2017.
2. J. Walker, "Industrial Uses of Drones - 5 Current Business Applications," *techemergence*, <https://www.techemergence.com/industrial-uses-of-drones-applications/>, 2018.
3. C. Joseph, "US Government Probes Airplane Vulnerabilities, Says Airline Hack Is Only a Matter of Time," *Motherboard*, <https://motherboard.vice.com/enus/article/d3kwzx/documents-us-government-hacking-planes-dhs>, 2018.
5. [4] W.Wei, "You Can Hijack Nearly Any Drone Mid Flight Using this Tiny Gadget", *TheHackerNews*, <https://thehackernews.com/2016/10/how-to-hack-drone.html>, 2016.
6. [5] Aircrack-ng:tools to assess WiFi network security, <https://www.aircrack-ng.org/>
7. A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned Aircraft Capture and Control via GPS Spoof-ing," *Journal of Field Robotics*, vol. 31, no. 4, pp.617-636, 2014.<https://dx.doi.org/10.1002/rob.21513>
8. A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned Aircraft Capture and Control via GPS Spoof-ing," *Journal of Field Robotics*, vol. 31, no. 4, pp.617-636, 2014.<https://dx.doi.org/10.1002/rob.21513>
9. V. Dey, V. Pudi, C. Anupam, and Y. Elovici, "Security Vulnerabilities of Unmanned Aerial Vehicles and Countermeasures: An Experimental Study," *VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID), 2018 31st International Conference on. IEEE*, pp.398-403, 2018, Available: <https://ieeexplore.ieee.org/abstract/document/8326960/>
10. "How To Get Source Code (JAVA&XML) From An Android APK File," *Techies Net*, www.cyberfreewishes.com/blog/how-to-get-source-code-java-xml-from-an-android-apk-file#.W1pPFNJKi5e
11. "HackRF One," *Great Scott Gadgets*, <https://greatscottgadgets.com/hackrf/>



Defense Techniques Against Cyber Attacks on UAV

Charan Gudla

Md. Shohel Rana

Andrew H. Sung



ABOUT THE AUTHOR

CHARAN GUDLA

PhD student, The University of Southern Mississippi



ABOUT THE AUTHOR

MD. SHOHEL RANA

Md. Shohel Rana is currently working as a Graduate Research Assistant and doing Ph.D. in Computational Science under the School of Computing Sciences and Computer Engineering at the University of Southern Mississippi. Md. does research in Digital Image Processing and Computer Vision, E-Learning, Web Technologies, Machine Learning, Distributed Databases, Cybersecurity

To find myself in a challenging job situation where in, I can contribute significantly to the Organization, which can catalyze my development in professional, personal and financial fronts.



ABOUT THE AUTHOR

ANDREW H. SUNG

Director and Professor, School of Computing at University of Southern
Mississippi

Unmanned aerial vehicles (UAVs) or drones serve a wide range of applications from surveillance to combat missions. UAVs carry, collect, or communicate sensitive information, which becomes a target for the attacks. Securing the communication network between the operator and the UAV is therefore crucial. So far, the networks used in most UAV applications are static, which allows more time and opportunity for the adversary to perform cyber-attacks on the UAV. In this article, we propose to study Moving Target Defense (MTD) techniques against cyber-attacks on drones including wireless network encryption and intrusion detection system. MTD techniques change the static nature of the systems to increase both the difficulty and the cost (effort, time, and resources) of mounting attacks. For illustration purposes, a well-known cyber-attack is performed on a popular commercial drone and results are presented to show the network vulnerabilities, damages caused due to the attacks and defense techniques to prevent the attacks.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) or drones are widely increasing in population [1]. Due to the fact that they are efficient, low cost, lightweight and easy to control, drones serve in applications such as military [2], monitoring [3] [4], disaster relief [5] and rescue operations [6]. UAVs are used to extend the wireless network coverage in the telecommunications field [7]. Amazon prime air [8] is a future service by Amazon that uses drones to deliver packages.

Though there are many advantages of drones, they are prone to various physical and cyber-attacks. The common forms of communication over a network to send and receive data are Satellite, Cellular, Wi-Fi, GPS, ZigBee. In 2009, Iraqi insurgents hacked predator drone feeds [9]. In 2011, a computer virus infected networks used by pilots controlling US Air Force drones at Creech Air Force Base in Nevada [10]. In 2011, an American Lockheed Martin RQ-170 Sentinel unmanned aerial vehicle was captured by the cyberwarfare unit of Iranian forces in Iran [11]. The predator drone video feeds were exposed online without the knowledge of the operator [12]. Wireless network jammers and GPS spoofing devices available at low cost are used to perform such attacks.

In this research, we discuss various vulnerabilities of UAVs and the hacking techniques are explored. Existing defense techniques that help in defense against cyber-attacks are reviewed. To show the vulnerabilities of the drone and exploitation, we created a base station and a well-known hacking technique is implemented on UAV Parrot AR Drone. By implementing hacking techniques, it shows that the attacker can cause severe damage to the drone or take control over it by compromising the wireless network between the operator and drone. The experiment helps to understand the importance of securing UAV systems against cyber-attacks.

The rest of the article is organized as follows. Section 2 introduces the related work done to prevent cyber-attacks on drones. Section 3 presents various hacking techniques that can be implemented to crash or take control over the drone. Section 4 demonstrates a hacking technique experiment implemented on the AR Drone. In Section 5 we discuss defense techniques against cyber-attacks on drones and the results are elaborated.

2. Related work

Various defense techniques have been proposed against these attacks on drones. In [13] Nils Miro Rodday et al. suggested the use of secure encryption schemes for Wi-Fi access points. In [14] Johann Pleban et al. showed a method of encrypting the wireless network where the drone acts as a client and the RC as an access point. The open Wi-Fi network is encrypted by WPA_supplicant to stop the attacker hacking the drone. In [15] Chaitanya Rani et al. illustrated vulnerabilities of drones and suggested encryption, intrusion detection systems as defense mechanisms. Kim Hartmann and Christoph Steup [16] developed a risk assessment scheme on services and communication infrastructure. James Goppert et al. [17] evaluated cyber-attack severity by establishing a metric to indicate the time of complete failure of the system. In [18] Robert Mitchell and Ing-Ray Chen developed a behavior rule-based UAV intrusion detection system for capturing malicious behavior when a UAV is under attack and prohibits its continuation.

3. Hacking techniques

In this section, UAV wireless network attacking techniques are discussed. Our experiment of hacking is applied on most popular drones and results are illustrated. Drone wireless networks can be hacked when the attacker knows the MAC address of a specific drone he wants to hack. The type of attacks on wireless networks of drones are as follows:

- Data packet capture
- Denial of service (DoS) attack
- Man-in-the-middle attack (MIMA)

3.1 Data packet capture

The hacker gathers the required information about the target by data packet capture method. The wireless network of the drone sends out the beacon frames that can be captured, and they consist of MAC addresses of the drone and remote-control device operating the drone, the type of encryption (WEP/WPA/WPA2/OPN) and the wireless network channel it is operating on. Aircrack-ng and Wireshark are the tools used to capture the wireless network frames.

3.2 Denial of service (DoS) attack

The wireless network [19] access points are hacked by deauthentication flood attacks (DoS) [20]. Continuous deauthentication requests are sent to the targeted access point exhausting its memory. Due to this, the clients cannot contact the access point since there is no memory left to reconnect with the clients, which leaves no connection between them. The deauthentication attack will target the MAC address of the access point, called as BSSID (captured from data packet capture), so that all the clients are disconnected from the access point, or use the MAC address of a

specific targeted client so it is disconnected. The clients try to reconnect with the access point, but they will fail until the deauthentication attack is stopped.

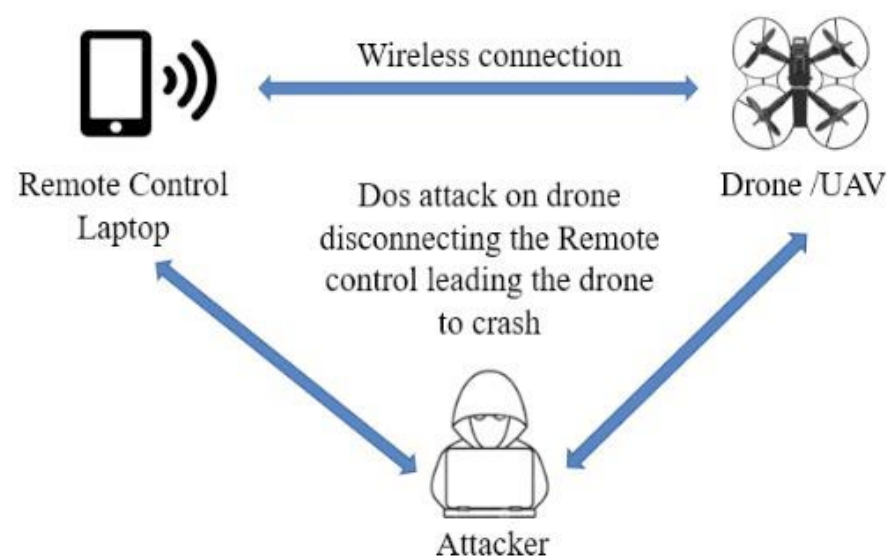


Fig. 1. Denial of Service (DoS) attack

3.3 Man-in-the-middle attack

The attacker spoofs and gains control over the communication network between the drone and remote control (RC) device user. The system details gathered from the initial data capture helps him send the authentication commands to the drone as if he is the original RC user. The data feed location from the drone will be seen by the hacker without the knowledge of both drone and RC user. If the wireless network is protected with a password, then by handshake protocol, the keys for authentication can be obtained by Aircrack-ng and crunch tools.

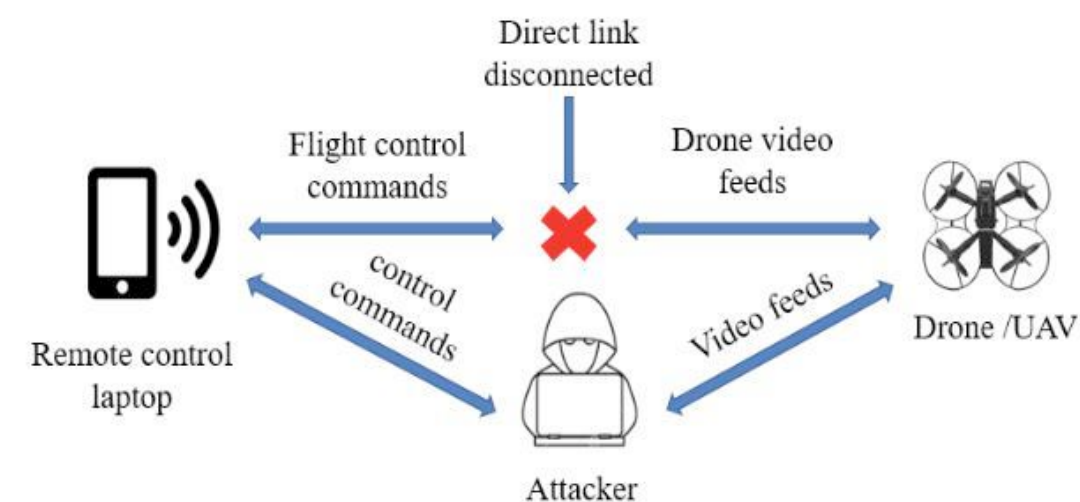


Fig. 2. Man-in-the-Middle Attack

4. Cyber-attack on drones

One of the most popular hacking techniques (DoS) is implemented on a drone's static network. We used a Parrot A.R drone for the experiment. In this technique, the remote control device is disconnected from the drone by continuously sending the deauthentication commands. The drone will crash immediately, or the attacker will take control of the drone by connecting to his device. Kali Linux in a virtual machine is used with a bridge adapter Alfa AWUS036NHA

USB wireless adapter. Aircrack-ng [21] is the suite containing the necessary tools to attack the drone. The following are the commands used to attack the drone.

```
root@kali: ~# iwconfig wlan0 mode monitor
```

```
root@kali: ~# ifconfig up
```

```
root@kali: ~# aireplay-ng -9 wlan0
```

```
root@kali: ~# airodump-ng wlan0
```

CH 3][Elapsed: 42 s][2018-03-03 19:38

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
A0:14:3D:F8:07:0B	-35	21	420	0	1	54e.	OPN		ardrone2
00:11:A3:1C:07:FA	-56	26	1	0	6	54	WPA2 TKIP	PSK	
00:02:6F:E6:1D:04	-81	15	70	0	11	54e.	WPA CCMP	PSK	
60:02:92:E0:7A:18	-90	1	0	0	1	54e.	WPA2 CCMP	PSK	CBCI-B827-2.4
60:02:92:E0:7A:1B	-90	0	0	0	1	54e.	WPA2 CCMP	MGT	XFINITY
60:02:92:E0:7A:1A	-91	1	0	0	1	54e.	OPN		xfinitywifi

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
(not associated)	B4:F7:A1:D0:9B:80	-69	0 - 1	21	3	
(not associated)	5C:AF:06:62:8F:FE	-91	0 - 1	0	1	
A0:14:3D:F8:07:0B	08:E6:89:35:F4:04	-25	0e- 0e	0	421	
00:11:A3:1C:07:FA	00:1E:04:FA:0D:BB	-1	54 - 0	0	1	

Fig. 3. Data packet capture showing MAC addresses

Executing the above commands will implement the data capture attack on a wireless network resulting in capturing of beacon frames consisting of source and destination MAC addresses. The MAC addresses shown in Fig. 3 are the drone's MAC address and the remote-control device listed as the station controlling the drone.

```
root@kali: ~# aireplay-ng -0 0 -a droneBSSID -c remotecontrolBSSID wlan0
```

The above command launches the cyber-attack on the drone leading it to crash. Fig. 4 and Fig. 5 show the communication link before and after the cyber-attack, respectively.



Fig. 4. Communication link before DoS attack

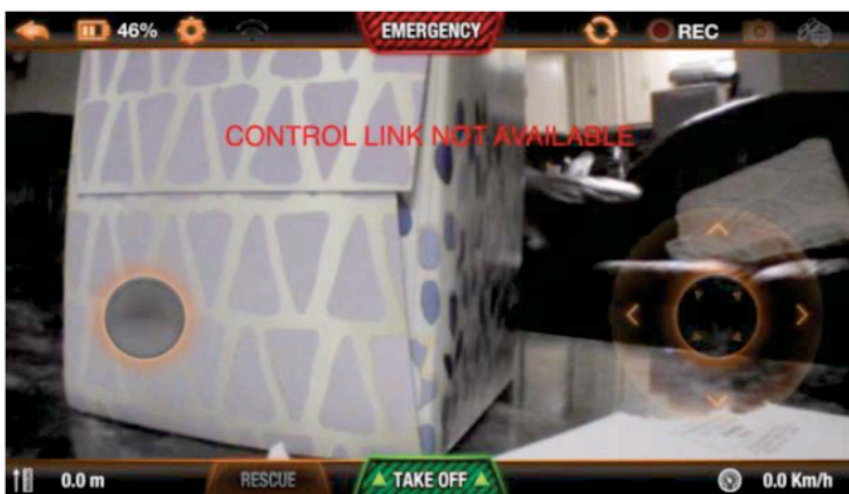


Fig. 5. Communication link after DoS attack

5. Defense against cyber attacks

For enhancing the security of the drones, we propose various defense techniques listed below.

- Wireless network encryption
- Intrusion detection system (IDS)
- Moving target defense (MTD)

We created a base station control system for a Parrot A.R drone that consists of the above security measures. The base station model is shown in Fig. 6 and Fig. 7.

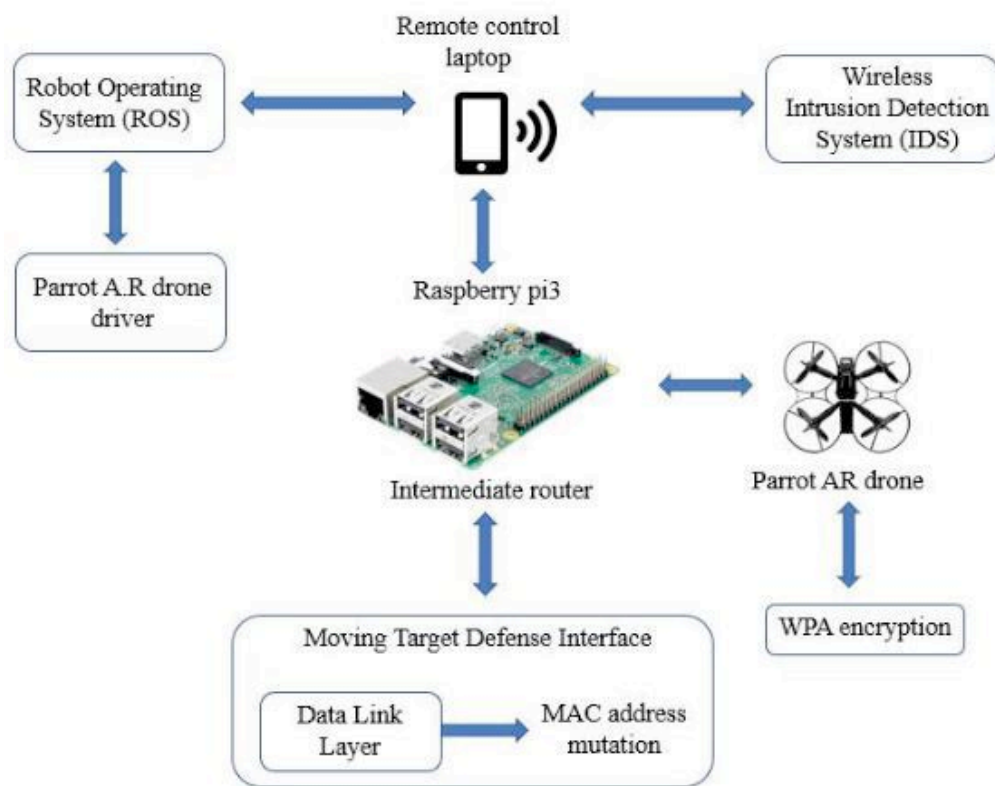


Fig. 6. Base station control system model

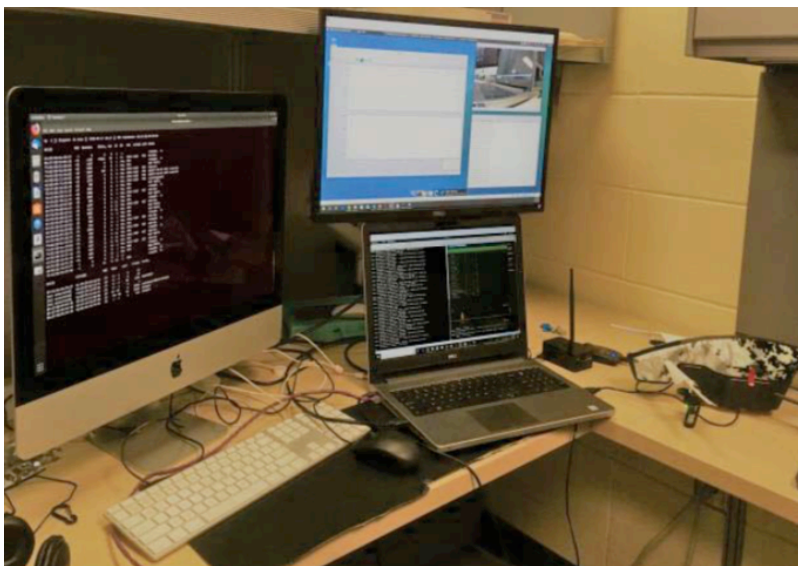


Fig. 7. Base station

Raspberry Pi is an affordable low-cost computer that can be used in different projects. We are using it as an intermediate router [22] and establish a secure wireless network between remote control and the drone. It is configured in such a way that it will act as a hotspot connecting devices into the network and make a communication link between them. The remote-control laptop sends control commands to the drone via Raspberry Pi router and the drone sends live video feed to the laptop through the Raspberry Pi router. The Raspberry Pi wireless network is secured with WPA2 encryption.



Fig. 8. Raspberry Pi

The Robot Operating System (ROS) [23] is a collection of tools and libraries that simplify the task of creating robust robotic applications. Part of this ROS consists of an AR drone driver to communicate with the drone and control it. Using ROS, we can develop autonomous tasks for the drone to accomplish.

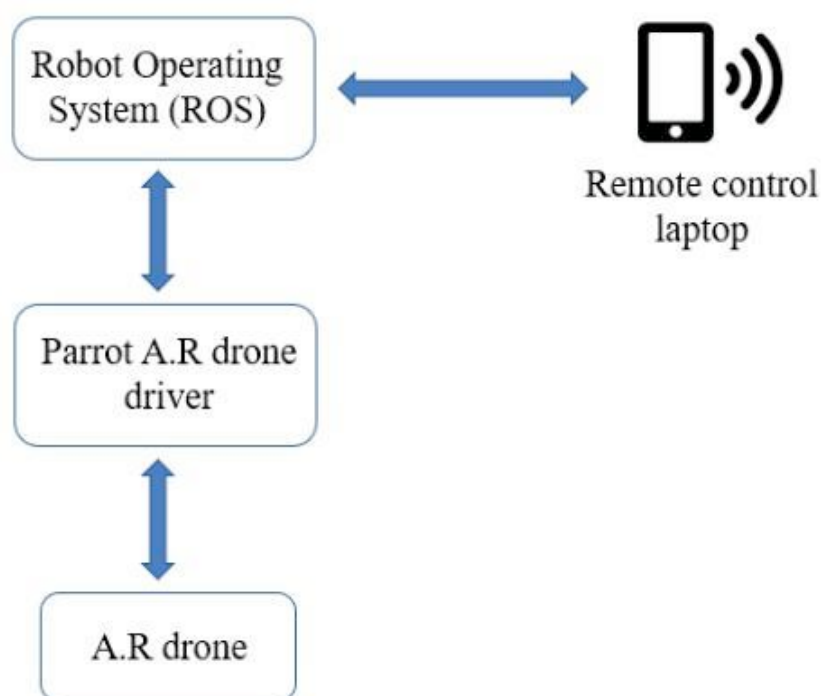


Fig. 9. ROS model

5.1 Wireless network encryption

Since, an AR drone will act as an access point and its network is unencrypted and open, multiple devices can be connected to it but only one device can control it. Disconnecting the authentic user and reconnecting to the drone with a fake user compromises the drone. The wireless network of the drone can be encrypted with WPA2 security by installing the compiled binaries of WPA_supplicant [24] into the drone libraries. The binaries `wpa_cli`, `wpa_passphrase`, `wpa_supplicant` should be included in the `bin` folder of the drone in order to accomplish it. After successful installation of the binaries, the drone will stop acting as an access point and it will connect to the provided access point name and passphrase (in our case, it will connect to the Raspberry Pi).

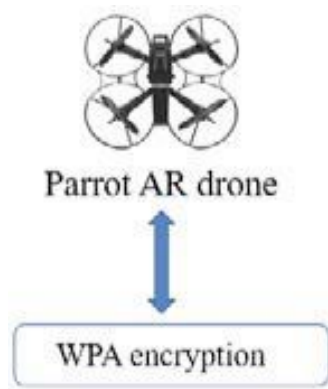


Fig. 10. WPA binaries in AR drone

5.2 Intrusion detection system

Intrusion detection systems monitor the wireless network in real time. Intrusion is an unauthorized entry into the network without the knowledge of the true owner. The systems can be spoofed, leading to direct access to the malicious user. The supervision of malicious activities, attacks, and spoofing on the network is intrusion detection.

An IDS is a kind of defensive tool but it doesn't provide preventive actions against the attacks. It's usually software that monitors the network behavior and notifies the user if there are any anomalies.

Kismet wireless IDS is used to monitor the drone wireless network [25]. The list of alerts is included in the Kismet configuration file to actively monitor the network and notify in case of any suspicious activities.

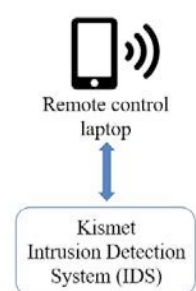


Fig. 11. Kismet IDS

5.3 Moving target defense

Moving Target Defense [26] is a technique where system characteristics are changed from static to dynamic, thus increasing the complexity for the hacker to attack. The network between any two nodes is considered static until now. This gives ample time for the attacker to gather the information regarding system configuration like OS, Network IP address, MAC address, etc. The information gathered is sufficient for the attacker to exploit vulnerabilities and launch attacks on the network. Moving target defense techniques completely change the game by implementing randomness in the system configuration, which makes it less static, less deterministic and less homogenous [26]. This makes the attacker spend more time, thus increasing the operational cost and complexity in understanding.

We used a Raspberry Pi to implement the moving target defense by changing the MAC address periodically. Fig. 12 shows the moving target defense model.

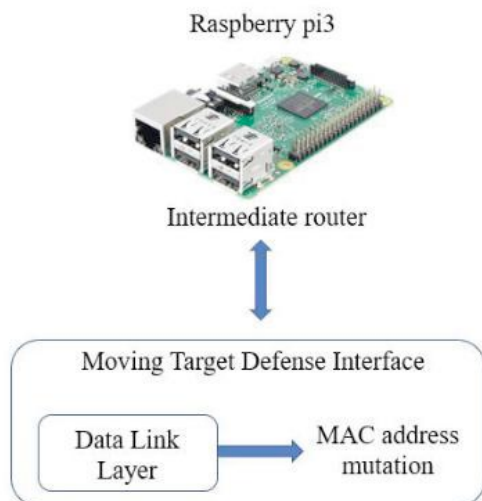


Fig. 12. Moving Target Defense model

6. Configuration

The following changes are made to the “kismet.conf” file to detect and monitor the wireless network. The following alerts will be generated in case of corresponding malicious activities on the network.

```
#kismet.conf
```

```
alert=NETSTUMBLER,10/min,1/sec
```

```
alert=WELLENREITER,10/min,1/sec
```

```
alert=LUCENTTEST,10/min,1/sec
```

```
alert=DEAUTHFLOOD,10/min,2/sec
```

```
alert=BCASTDISCON,10/min,2/sec
```

```
alert=CHANCHANGE,5/min,1/sec
```

```
alert=AIRJACKSSID,5/min,1/sec
```

```
alert=PROBENOJOIN,10/min,1/sec
```

```
alert=DISASSOCTRAFFIC,10/min,1/sec
```

```
alert=NULLPROBERESP,10/min,1/sec
```

```
alert=BSSTIMESTAMP,10/min,1/sec
```

```
alert=MSFBCOMSSID,10/min,1/sec
```

```
alert=LONGSSID,10/min,1/sec
```

```
alert=MSFDLINKRATE,10/min,1/sec
```

```
alert=MSFNETGEARBEACON,10/min,1/sec  alert=DISCONCODEINVALID,10/min,1/sec
```

```
alert=DEAUTHCODEINVALID,10/min,1/sec
```

```
#Do we have a GPS?
```

```
gps=fals
```

```
#Log file directory
```

```
configdir=/var/log/kismet/
```

MAC address mutation in layer 2 of the OSI model is accomplished using the compiled libraries of macchanger tool in the Raspberry Pi by executing the following script.

```
#!/bin/bash
```

```
macchanger --show wlan0
```

```
Ifconfig wlan0 down
```

```
macchanger -r -b wlan0
```

```
Ifconfig wlan0 up
```

```
macchanger --show wlan0
```

```
sudo service network-manager start
```

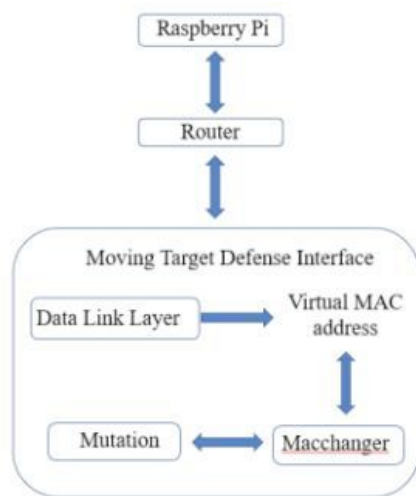



Fig. 13. MAC address mutation model

7. Results

In this section, we show the implementation of defense techniques against cyber-attacks on a Parrot AR drone wireless network and alerts produced by an IDS, which is monitoring the drone wireless network. The data capture attack gathers the required information about the network to launch the attack as shown in Fig. 14. The wireless network is encrypted with WPA2 security, which can be seen under the ENC column in Fig. 14 so that the attacker cannot directly connect or intercept the wireless network.

```
CH 2 ][ Elapsed: 36 s ][ 2018-06-06 20:01
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
B8:27:EB:84:B4:3B	-33	21	77 0	1	54	WPA2	CCMP	PSK	hotspot

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
B8:27:EB:84:B4:3B	A0:14:3D:F8:07:0B	-20	0 - 0e	49	7	
B8:27:EB:84:B4:3B	00:1E:64:FA:0D:BB	-28	54 - 6e	0	6	

Fig. 14. Data capture attack

```
root@kali: ~# aireplay-ng -0 0 -a droneBSSID -c remotecontrolBSSID wlan0
```

Using the above command, a DoS attack can be implemented on the drone wireless network. Since the drone and remote-control laptop are connected to a Raspberry Pi, the cyber-attack will be launched on the MAC address of the Raspberry Pi. The wireless network is named “hotspot” as shown in Fig. 14. The associated drone and remote-control laptop MAC addresses are listed under the STATION column.

Since, the wireless network is monitored by the Kismet IDS, it will detect and alert the user about the malicious activity on the network as shown below in Fig. 15.

```
INFO: Detected new managed network "hotspot", BSSID B8:27:EB:84:B4:3B,
encryption yes, channel 1, 72.20 mbit
ALERT: BCASTDISCON Network BSSID B8:27:EB:84:B4:3B broadcast deauthenticate
/disassociation of all clients, possible DoS
ALERT: BCASTDISCON Network BSSID B8:27:EB:84:B4:3B broadcast deauthenticate
/disassociation of all clients, possible DoS
ALERT: BCASTDISCON Network BSSID B8:27:EB:84:B4:3B broadcast deauthenticate
/disassociation of all clients, possible DoS
ALERT: BCASTDISCON Network BSSID B8:27:EB:84:B4:3B broadcast deauthenticate
/disassociation of all clients, possible DoS
ALERT: BCASTDISCON Network BSSID B8:27:EB:84:B4:3B broadcast deauthenticate
/disassociation of all clients, possible DoS
```

Fig. 15. Kismet IDS alerts

The moving target defense technique is implemented to mutate the MAC address of the Raspberry Pi. Now that the MAC of the Raspberry Pi is changed, the cyber-attack will fail because the attack is launched on the previous MAC address. The MAC address of the Raspberry Pi after the random mutation is shown in Fig. 16 detected by the Kismet IDS.

```
INFO: Detected new managed network "hotspot", BSSID FE:DA:B4:38:B3:EE,
encryption yes, channel 1, 72.20 mbit
```

Fig. 16. New MAC address after mutation

The mutation of the MAC address makes it possible for Kismet to detect the wireless network with the same domain name but with a different MAC as shown in Fig. 17.

```
INFO: Detected new managed network "hotspot", BSSID 2E:5F:6C:5B:2D:EE,
encryption yes, channel 1, 72.20 mbit
```

Fig. 17. New MAC address after mutation

When the hacker attacks the wireless network with the same initial MAC address without the knowledge that the MAC address is changed, the deployed hacking technique will fail to engage as shown in Fig. 18 saying no such BSSID available.

```
root@kali:~# sudo aireplay-ng -0 0 -a B8:27:EB:84:B4:3B wlan0mon0
23:50:41 Waiting for beacon frame (BSSID: B8:27:EB:84:B4:3B) on channel 1
23:50:51 No such BSSID available.
Please specify an ESSID (-e).
```

Fig. 18. Failure of cyber-attack

The navigational data transmitted from the drone to the base station contains the acceleration, velocity, altitude and the four motors' rotational speeds as shown in Fig. 19.

```
Terminal - ardrone@ardrone-vm: ~
File Edit View Terminal Go Help
rotY: 0.51700001955
rotZ: 17.6319999695
altd: 0
vx: 9.1018409729
vy: 32.0157852173
vz: -0.0
ax: -0.0387297905982
ay: 0.0369769856334
az: 0.963544905186
motor1: 0
motor2: 0
motor3: 0
motor4: 0
tags_count: 0
tags_type: []
tags_xc: []
tags_yc: []
tags_width: []
tags_height: []
tags_orientation: []
tags_distance: []
tm: 1164184.0
```

Fig. 19. Navigational data from the drone

The navigational data from the drone contains acceleration and estimated velocity values, plotted as shown in Fig. 20.

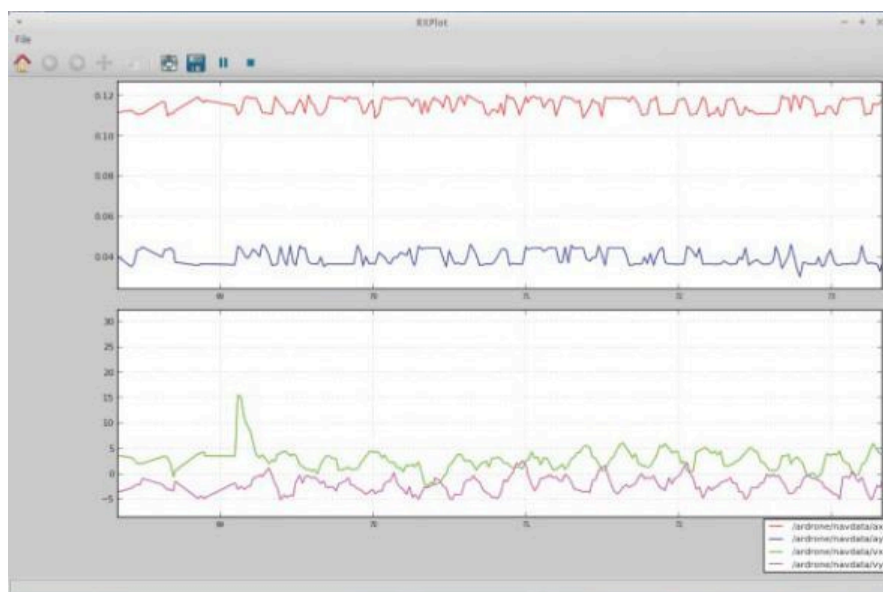


Fig. 20. Acceleration and velocity plots

8. Conclusion

The commercial, civilian, and military applications of UAVs are increasing rapidly, and the vulnerabilities of UAVs create risks to public and private sectors as drones can carry payloads as well as sensitive information, picture, and video feeds, etc. To mitigate the risk, we need to analyze the various vulnerabilities and attack techniques and develop defense techniques for drones against physical and cyber-attacks.

Cyber-attacks on UAVs can easily exploit the static nature of wireless networks connecting remote control devices and UAVs. The experiments reported in this article illustrate various vulnerabilities of the network that can be exploited to crash the drone or to take over its control. By implementing wireless encryption, intrusion detection systems, or MTD techniques, the system becomes more complex for the attacker to exploit any vulnerabilities and implement/launch

attacks. Even though the attacker collects the required information to implement a cyber-attack, the network characteristics will be changed, and the attack will fail to engage or execute. In this way, the wireless network is hardened to protect the drones against different cyber-attacks.

In addition to the military, homeland security organizations are also interested in R&D on moving target defense techniques [27], as the name UAV suggests, unmanned aerial vehicles can accomplish a wide range of missions without the high cost or risks of manned flights. Thwarting cyber-attacks on drones is therefore critical for successful deployment of UAVs and a comprehensive study, implementation, analysis and evaluation of MTD techniques outlines the scope of future work of this project. Application of protected management frames (PMF) service to the network will also defend against cyber-attacks. Future work will study software and platform based moving target defense techniques for drones.

References

1. “FAA estimates 7 million drones by 2020”,
<https://gcn.com/articles/2016/03/28/faa-drone-projections.aspx> (28 March 2016,
accessed 06 June 2018)
2. Udeanu, Gheorghe, et al. “Unmanned Aerial Vehicle in Military Operations.” Scientific Research and Education in the Air Force, vol. 18, no. 1, 2016, pp. 199-206.,
doi:10.19062/2247-3173.2016.18.1.26
3. Kafi, Mohamed Amine, et al. “A Study of Wireless Sensor Networks for Urban Traffic Monitoring: Applications and Architectures.” Procedia Computer Science, vol. 19, 2013, pp. 617–626., doi: 10.1016/j.procs.2013.06.082
4. Alvear, Oscar, et al. “Using UAV-Based Systems to Monitor Air Pollution in Areas with Poor Accessibility.” Journal of Advanced Transportation, vol. 2017, 2017, pp. 1– 14.,
doi:10.1155/2017/8204353
5. Debusk, Wesley. “Unmanned Aerial Vehicle Systems for Disaster Relief: Tornado Alley.” AIAA Infotech@Aerospace 2010, 2010, doi:10.2514/6.2010-3506
6. Waharte, Sonia, and Niki Trigoni. “Supporting Search and Rescue Operations with UAVs.” 2010 International Conference on Emerging Security Technologies, 2010,
doi:10.1109/est.2010.31
7. Guillen-Perez, Antonio, et al. “Wi-Fi Networks on Drones”. 2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT), 2016, doi:10.1109/itu-wt.2016.7805730
8. Amazon prime air delivery using drones to deliver the ordered packages,
<https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011> (accessed 06 June 2018)
9. Iraqi insurgents hacked predator drone feeds,
<http://www.cnn.com/2009/US/12/17/drone.video.hacked/index.html> (17 December 2009,
accessed 18 July 2018)
10. Computer virus infects drone plane command centre US,
<https://www.theguardian.com/technology/2011/oct/09/virus-infects-drone-plane-command>
[d](https://www.theguardian.com/technology/2011/oct/09/virus-infects-drone-plane-command) (9 Oct 2011, accessed 18 July 2018)
11. American Lockheed Martin RQ-170 Sentinel unmanned aerial vehicle (UAV) was captured by Iranian forces, https://en.wikipedia.org/wiki/Iran%E2%80%93U.S._RQ-170_incident
(accessed 18 July 2018)

References

12. Predator drone video feeds exposed online, <https://www.bleepingcomputer.com/news/government/us-government-leaves-predator-drone-video-feeds-exposed-online/> (05 May 2015, accessed 18 July 2018)
13. N. M. Rodday, R. D. O. Schmidt, A. Pras. "Exploring security vulnerabilities of unmanned aerial vehicles", NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, pp. 993-994, Apr 2016
14. J. S. Pleban, R. Band, R. Creutzburg, R. Creutzburg, D. Akopian, "Hacking and securing the AR.Drone 2.0 quadcopter: Investigations for improving the security of a toy", International Society for Optics and Photonics, pp. 90300L, feb 2014
15. Rani C, Modares H, Sriram R, Mikulski D, Lewis FL (2016): Security of unmanned aerial vehicle systems against cyber-physical attacks. Journal of Defense Modeling and Simulation: Applications, Methodology, Technology 2016, Vol. 13(3) 331–342 The Author(s) 2015 DOI: 10.1177/1548512915617252
16. K. Hartmann, C. Steup, "The vulnerability of UAVs to cyber-attacks an approach to the risk assessment", Cyber Conflict (CyCon) 2013 5th International Conference on, pp. 1-23, 2013
17. Goppert, James, et al. "Numerical Analysis of Cyberattacks on Unmanned Aerial Systems." Infotech@Aerospace 2012, 2012, doi:10.2514/6.2012-2437
18. R. Mitchell, I.-R. Chen, "Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications", IEEE Trans. Syst. Man Cybern. Syst., vol. 44, no. 5, pp. 593-604, May 2014
19. "Wi-Fi.", Microchip Developer Help, <http://microchipdeveloper.com/wifi:start> (accessed 12 May 2018)
20. Compton, Stuart: 802.11 Denial of Service Attacks and Mitigation, SANS Institute InfoSec Reading Room
21. Aircrack-ng, <https://www.aircrack-ng.org/> (accessed 17 June 2018)
22. RPI-Wireless-Hotspot for raspberry pi to convert into router, <https://github.com/harryallerston/RPI-Wireless-Hotspot> (accessed 18 June 2018)
23. Robot Operating System, <http://www.ros.org/about-ros/> (accessed 16 June 2018)

References

24. WPA2 encryption, <https://github.com/daraosn/ardrone-wpa2> (accessed 16 June 2018)
25. Kismet wireless intrusion detection system for drone, <https://raw.githubusercontent.com/kismetwireless/kismet/master/README> (accessed 16 June 2018)
26. H. Okhravi, M.A. Rabe et al., "Survey of Cyber Moving Targets", Lincoln Laboratory - Massachusetts Institute of Technology Technical Report, September 2013
27. "Moving Target Defense", Homeland Security. <https://www.dhs.gov/science-and-technology/csd-mtd> (accessed 15 June 2018)



Security Analysis of FHSS-type Drone Controller

Kibum Choi

Youngseok Park



ABOUT THE AUTHOR

KIBUM CHOI

School of Electrical Engineering, KAIST, Daejeon, Republic of Korea



ABOUT THE AUTHOR

YOUNGSEOK PARK

Graduate School of Information Security, KAIST, Daejeon, Republic of
Korea

Unmanned Aerial Vehicles (UAVs), or drones, have attracted a considerable amount of attention due to their utility to civilians as well as military applications. However, the security issues involved in UAV technology have not been extensively discussed in the literature. As a first step toward analyzing these security issues, we investigate security in drone controllers, especially controllers that adopt Frequency Hopping Spread Spectrum (FHSS). In order to affect an FHSS-type controller, an attacker first has to access its physical layer. This is difficult because of the pseudorandomness of the hopping sequence and the rapidly changing channels. However, these difficulties can be relaxed when the attacker acquires the hopping sequence and when the hopping speed of the target system is not significant. In this article, we propose a general scheme to extract the hopping sequence of FHSS-type controllers using a software-defined radio (SDR). We also propose a method to address the issue of the limited bandwidth of the SDR. We implemented our scheme on a Universal Software Radio Peripheral (USRP), successfully extracted the hopping sequence of the target system, and exposed the baseband signal.

1. INTRODUCTION

Because of their extensive range of application, from military airstrikes [20] to automated package delivery platforms [2, 4], unmanned aerial vehicles (UAVs) or drones have lately been the subject of increasing interest. As they become more popular, drones are frequently flown in noisy environments, and are thus exposed to intentional/unintentional interference. It is therefore necessary for drones to secure their control systems against such interference. As exemplified by the case of the capture by Iranian forces of a United States Air Force (USAF) drone in 2011 [21], even military drone control systems are not adequately secure.

Wireless remote controllers for radio-controlled (RC) model aircraft in the past employed fixed frequencies of tens of megahertz [1]. However, due to a shortage of spectral resources, and in order to protect against interference, current remote controllers adopt spread spectrum technology on industrial, scientific, and medical (ISM) radio bands of 2.4 GHz [1, 11].

FHSS is a spread spectrum technology that continuously changes carrier frequency for anti-jamming/sniffing/spoofing capabilities. A theoretically unique hopping sequence is pre-shared by every transmitter-receiver pair through binding, which can prevent issues of mutual interference. However, even FHSS cannot completely protect links against all jamming/sniffing/spoofing threats. High-energy wideband jammers can block the entire hopping space used by an FHSS system [17]. A random jammer with a much greater hopping speed can deteriorate the signal-to-noise ratio (SNR) [17]. Furthermore, FHSS is vulnerable to reactive jammers with a sufficiently high reaction speed.

Although the above-mentioned attack vectors against FHSS are quite expensive for attackers, an exposed hopping sequence can drastically reduce the required complexity of attacks. Using the hopping sequence, attackers can proactively react to the changing center frequency, which enables the implementation of low-cost reactive jammers or baseband extractors.

In this article, we propose a general scheme to extract the hopping sequences of FHSS-type drone controllers by using a software-defined radio (SDR). The versatility of SDRs makes it possible to deal with most FHSS-type drone controllers, which are mostly incompatible with one another [5]. We also propose a method to overcome the problem of limited SDR bandwidth when treating controllers with larger bandwidths. We applied our proposed scheme for a real-world FHSS-type controller, where the bandwidth of the target controller was approximately three times larger than that of the SDR, successfully extracted the total hopping sequence of the target system, and exposed the baseband signal.

The remainder of this paper is structured as follows: Sec. 2 provides the requisite background for our research here, whereas Sec. 3 is dedicated to a description of our proposed scheme to extract hopping sequences. In Sec. 4, we describe the implementation of the attack platform as well as the results. Sec. 5 summarizes related work in the area, Sec. 6 discusses future works and Sec. 7 contains our conclusions.

2. Background and Attack Model

2.1 Frequency Hopping Spread Spectrum

FHSS is a major spread spectrum technology along with Direct Sequence Spread Spectrum (DSSS). In wireless communications, it rapidly switches channels using a pseudorandom sequence, which makes it difficult to eavesdrop. Rapidly changing carrier frequency also renders the system highly resistant to narrow-band interference, and enables it to share the frequency band with other systems using different communication technologies.

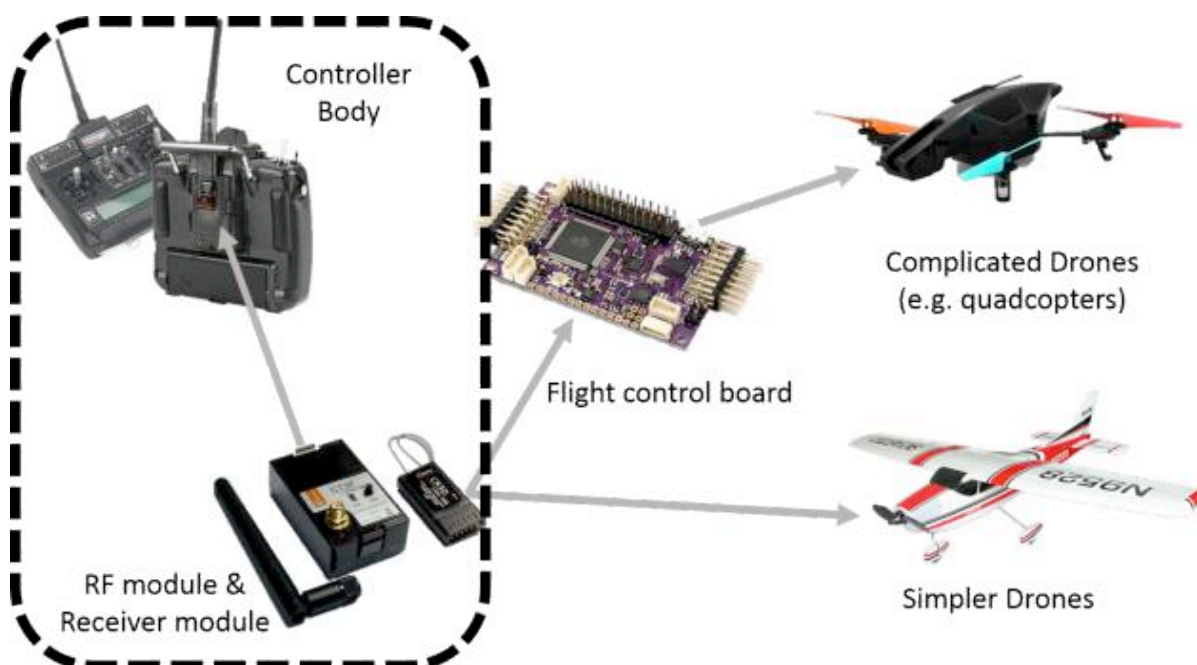


Fig. 1: RC aircraft system composition

FHSS has drawbacks as well. First, FHSS systems occupy much wider bandwidth than it actually requires. For example, a typical 10 channel FHSS system occupies bandwidth ten times wider than it actually uses. Second, a transmitter-receiver pair has to be finely synchronized, which is achieved several ways. A transmitter may transmit

duplications of a packet for all channels, while the receiver listens to a randomly selected channel. For another way, a transmitter-receiver pair can share a frequency table, repeating the predefined sequence.

Most drone controllers utilize the 2.4 GHz band, which is shared by many other wireless devices. Therefore, drone controllers can occupy only a fraction of assigned bandwidth. Furthermore, a transmitter-receiver pair does not have an additional communication channel. Once bound at the initial stage, the pair communicates with each other without any prior pairing steps after they are turned on. This means a consistent frequency table is shared by the pair.

2.2 *Radio control system for RC aircraft*

Drone controllers interface humans and drones. Although drones differ in their level of autonomy, controllers always take up the most critical functions, which makes controllers one of the most important components. Controllers vary as the drones are unique. In this paper, we focus on controllers for civilian RC aircraft.

A typical RC controller consists of three components: a transmitter body, an RF module, and a receiver. The transmitter body provides the user interface, and converts user control into electrical signals. The RF module modulates and upconverts the control signal. It characterizes the wireless link between a transmitter and a receiver, whose robustness against interference according to its wireless characteristics. The receiver reconverts the wireless signal into Pulse-width modulation (PWM) pulses. Fig. 1 shows the composition of the overall drone system, where components in the dashed rectangle indicate the drone controller.

Frequently, multiple RC aircraft are flown together, where multiple control signals interfere with one another. In this case, control signal interference is critical for drones. They can fall into an uncontrollable state, which is critical for fast moving aircraft. Therefore, RC controllers are required to resist high levels of interference. To both share the band and resist mutual interference, spread spectrum technologies (FHSS and DSSS) are widely adopted.

Currently, no industrial standard for RC controllers exists. Therefore, controllers from diverse manufacturers are usually not compatible. Furthermore, the absence of the standard makes manufacturers hide details of their products. This makes the RC controller a blackbox system for third party analysts.

2.3 *Attack Model*

Our attack model is as follows. First, the target system is considered to be a blackbox. Though the attacker can analyze the system on her own capabilities, she cannot access any confidential information on the system a priori. Second, we assume the controller signal has at least one exclusively distinguishing characteristic. Furthermore, the attacker can exploit this characteristic to differentiate the target signal on air from other signals. The attacker can easily purchase such popular controllers and analyze their signal to reveal exclusive characteristics.

3. Methodology

3.1 *Extracting the hopping sequence*

Measuring channel information

Typical FHSS systems have identical channel widths. Thus, we can derive channel center frequencies and the number of channels from measurements at the lowest and the highest channels. The center frequency of the remaining channels can be identified by repeatedly adding channel bandwidth to the first channel until the last channel is reached.

Detecting channel activeness

In order to extract the hopping sequence through measurement, we first need to detect channel activeness. A considerable amount of past research in the area has dealt with this topic, since it is related to cognitive radio [9, 18, 19, 23, 24]. Yucek et al. [24] listed various methods to determine spectral activeness according to the amount of available information regarding the target signal. The more accurate the method is, the more detailed the prior information that it requires.

Since the target system is considered to be a blackbox, applicable channel sensing methods are quite limited. Energy detection [19] and cyclostationarity-based detection [9] methods are representative techniques used to detect blackbox signals. We use energy detection to detect channel activeness for the sake of simplicity. However, this can be altered without affecting the remainder of our work here. Once we can detect channel activeness, we can record the history of the activated channels. Finally, assuming constant hopping speed, measurements of the continuous signal can be converted into a discretized sequence.

Searching the period

The easiest way to predict the future sequence is to extract the period in the hopping sequence. This can be achieved by choosing a part of the sequence and searching for repetitions. While this scheme works well in most cases, there are instances of error. If the length of the chosen part is greater than twice the actual period, the period appears longer than it is. Moreover, if the chosen part is shorter than half the actual period, the period can appear shorter than it is in some bad cases. These erroneous cases can be settled by searching repetitions by choosing multiple parts.

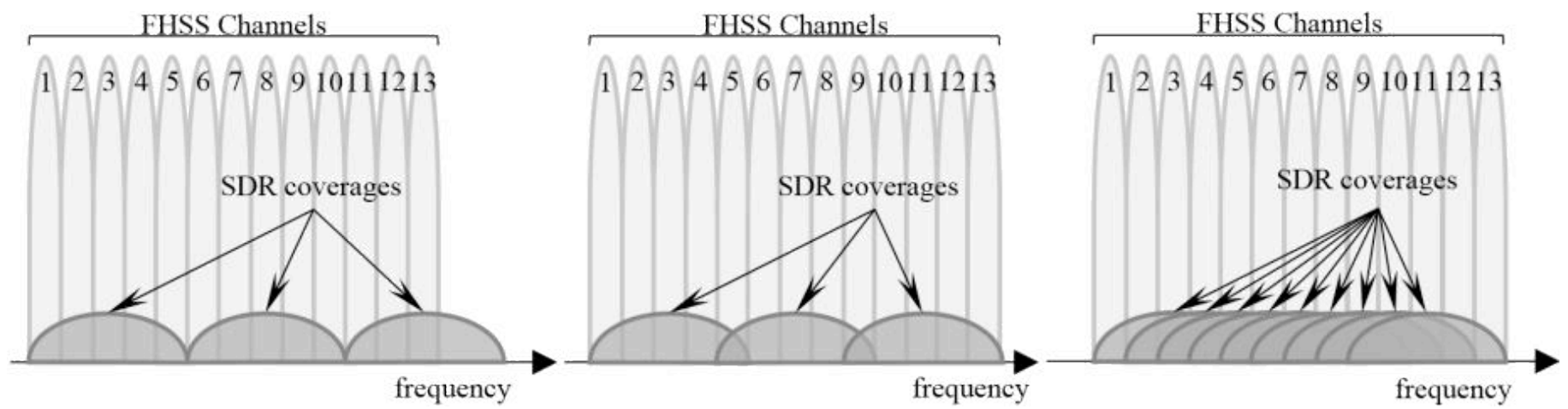


Fig. 2: Examples of SDR coverage arrangements

If the history has numerous measurement errors, the aforementioned exact matching-based search will fail. In such cases, we can find the period with similarity-based matching. This is identical to exact matching-based searches except that repetitions are detected by similarity scores. Whenever the similarity score exceeds a certain threshold during the search, the relevant points are marked as repeating points. If a sufficient number of points are acquired, the intervals between any two points are aligned and compared to identify the most frequently appearing interval, which can be considered the hopping period.

Various pattern matching algorithms can be used to derive similarity scores. Matched filters are largely adopted for pattern detection [7, 8]. Algorithms used to solve sequence alignment problems [22] can also be considered, since that problem is quite similar to the one here.

3.2 Overcoming limited SDR bandwidth

In some cases, the bandwidth of the target system can exceed the maximum bandwidth of the SDR used. In such cases, the SDR can only monitor a part of each channel. This makes it impossible to detect the activeness of channels beyond the tuned SDR bandwidth. To solve this issue, the attacker can simultaneously utilize multiple, tightly synchronized SDRs, or a more powerful SDR that can cover the entire range of the target bandwidth. However, these approaches are expensive. We suggest an alternative that enables a single narrowband SDR to acquire the full hopping sequence of the target system.

Measuring the number of channels and their center frequencies is not challenging, even with a narrowband SDR, since we simply need to measure the first and the last channels. However, we can only acquire a number of partial sequences with a narrowband SDR, and such partial sequences should be uniquely merged to obtain the actual total hopping sequence. In order to uniquely combine partial sequences, SDR coverage should be carefully arranged. We explore various arrangements to show that a careful arrangement can yield the total sequence without ambiguity. Note that in all examples of partial sequences presented in this subsection, all channels in a period are activated equally frequently. Although this condition is not essential to our method, typical FHSS systems meet this condition in order to uniformly utilize bandwidth.

Fig. 2 shows three examples of coverage arrangement. In the left arrangement, the SDR coverages span the entire hopping space but do not overlap with one another. In this arrangement, partial hopping sequences are combined only with the duration of slots of no activity, which can lead to multiple combinations. For example, if channels under each coverage are activated in a series: \1 5 2 3 4 / 10 6 9 8 7 / 13 11 12," the partial sequences for each coverage are \1 5 2 3 4," \10 6 9 8 7," and \13 11 12," respectively. Since the channels in each coverage are contiguously activated, these partial sequences can also be combined as, for example, \1 5 2 3 4 / 13 11 12 / 10 6 9 8 7." This leads to multiple combinations.

The middle arrangement can also lead to multiple solutions when each channel is activated more than once in a period. The two example hopping sequences below show one of such cases. Overlapped channels are marked with hats and differences are bolded. It is easily verified that the two hopping periods, (1) and (2), are different, although the corresponding partial sequences are identical.

$$\begin{array}{cccccccccccccccc} \cdots & 1 & \hat{5} & 2 & 3 & 4 & \mathbf{10} & 6 & \hat{9} & 8 & 7 & \mathbf{13} & \mathbf{11} & \mathbf{12} & \cdots & \\ \cdots & 1 & 5 & 2 & 3 & 4 & \mathbf{11} & 6 & 9 & 8 & 7 & \mathbf{10} & \mathbf{12} & \mathbf{13} & \cdots & \end{array} \quad \begin{array}{l} (1) \\ (2) \end{array}$$

By contrast, the last arrangement, which is maximally overlapped, does not lead to multiple solutions. In this arrangement, every channel, except the one at each end, overlaps with another, i.e., they are all entangled. Therefore, any rearrangement of channels different from the original will always interfere with other partial sequences. However, maximal overlap is not always optimal. In most cases, a loosely overlapped arrangement will suffice. Indeed, we can uniquely combine partial hopping sequences with a non-maximally overlapped arrangement. Therefore, repeated trials with increasing overlaps are required to find the optimally overlapped arrangement.

3.3 Possible attack vectors

Once the total hopping sequence has been acquired, the basic requirements of catching up the ongoing FHSS signal are met. With the hopping sequence of the target system in hand, the attack cost can be greatly reduced, and the hopping sequence can be applied to the following attack platforms.

Baseband extractors receive and record the baseband signal while continuously following the FHSS signal stream. The extracted baseband signal can later be analyzed to yield information regarding the modulation, encoding, or the packet structure of the baseband. Reactive sniffers operate similar to baseband extractors, except that they can demodulate and decode the baseband signal to expose the bitstream or meaningful information from target systems. Reactive jammers transmit narrowband interfering signals whenever a channel is activated. Using the extracted hopping sequence, the level of difficulty of implementing reactive jammers can be drastically reduced, since attackers can proactively wait for the channel to be activated.

4. Implementation and Results

4.1 Equipment

Software-defined Radio - We used a USRP N210 to receive signals from the target system. USRP N210 has a gigabit Ethernet interface, and can provide up to 25 million 16-bit pair (I & Q) samples per second ($= 2 \times 2B \times M = s = 100MB/s$) [6]. USRP N-series devices require a separate RF frontend, called a daughterboard. We used a CBX daughterboard [3] with full duplex capability with 40 MHz of instantaneous bandwidth. It can cover 1.2 6 GHz.



Fig. 3: FrSKY DJT Radio Telemetry (RF module, left of left gure), FlySky FH-TH9X Transmitter (transmitter body, right of left gure), and FrSKY D4R-II 4Ch Receiver (receiver, right gure)

Host PC - We used a general desktop with Intel Core i5-3570 and 16 GB of DDR3 memory. To interface with the USRP, we used Intel PRO/1000 PT Dual Port Server Adapter. As an OS, we used Ubuntu 12.04 LTS 64-bit.

4.2 Test Target Selection and Basic Analysis

Selected Test Target - We chose a real-world radio controller to verify our attack scheme. The target controller is composed of three components: a transmitter body, an RF module, and a receiver. The detailed brands and names are shown in Fig. 3. It adopts Advanced Continuous Channel Shifting Technology (ACCST), which is FrSKY's commercial name for FHSS. ACCST devices shift channels more than a hundred times per second for security and stability.

We first conducted basic examinations of the target system in order to apprehend its mechanism. We analyzed only the body and the RF module, since the analysis of the receiver is not required to verify the attack model.

Analysis of Transmitter Body. For the selected target, the transmitter body only output a series of PWM pulses and passed them to the RF module. The PWM pulses were further modulated and upconverted in the RF module. Having examined the transmitter body, we concluded that the transmitter body was not related to generating FHSS signals.

Analysis of the RF module The RF module was powered by a pin connecting it to the transmitter body, and modulated the input PWM pulses to generate the FHSS signal output. In order to analyze only the output signal of the RF module, we connected the module's output port and the CBX input port directly using a SubMiniature version A (SMA) cable. We then ran `uhd_fft` to view the spectrogram of the RF module's output signal. `uhd_fft` is a GUI application that

makes USRP work as a simple spectrometer. Since we already knew that the RF module used 2.4 GHz bands, we first tuned uhd_fft to 2.4 GHz, and gradually changed the frequency. As a result, the center frequency of the first and the last FHSS channel were found to be 2.40517 GHz and 2.41415 GHz, respectively.

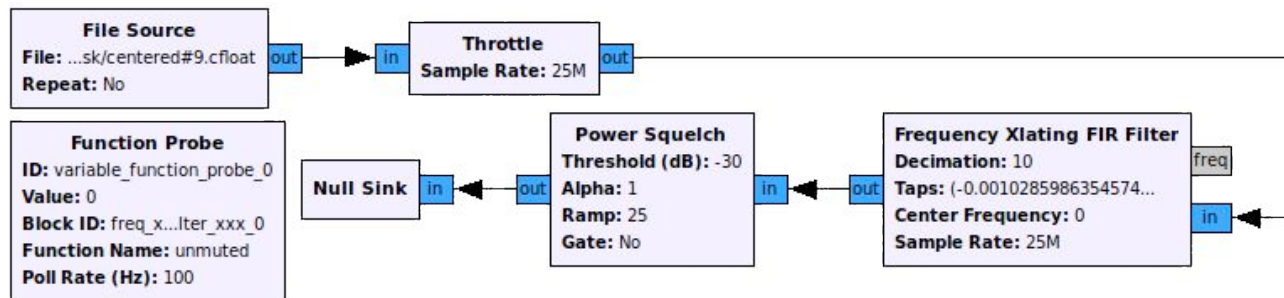


Fig. 4: GNURadio flow graph for partial sequence extraction

From these observations, we identified the total number of FHSS channels and their center frequencies.

To summarize, there were 47 channels in total, and each channel was 1.5 MHz wide. The total bandwidth was calculated as below.

$$\left[(2.40517 \times 10^9 - 1.5 \times 10^6 / 2) - (2.47415 \times 10^9 + 1.5 \times 10^6 / 2) \right] \text{ Hz} \approx 70 \text{ MHz}$$

It was approximately three times larger than the maximum bandwidth of USRP N210 (25 MHz), which was the case described in Section 3.2.

4.3 FHSS Sequence Extraction

Hopping Speed - The hopping speed of the target system is important because USRP has limited agility. If the hopping speed is too high, it is impossible to follow the changing frequency of the target system.

The simplest method of measuring hopping speed is to measure the duration of a hop, since typical FHSS systems have a constant hopping speed. To measure the duration, we first tuned the USRP to one of the channels and recorded the signal into a file. We subsequently browsed the recorded file to measure the duration of a hop, which was 0.0058 s. Converting this duration directly into hopping frequency, we derived $1 / 0.0058 \text{ s} = 172 \text{ Hops/s}$. Note that this was the upper bound of the hopping speed, since no FHSS channel is typically changed without a delay.

Based on work by Nychis et al. [13], the hopping speed was in a range that USRP can readily follow without any modifications to the field-programmable gate array (FPGA) or the firmware. In their study, the overall round trip time between the host and the USRP was measured to be 612 s on average with a standard deviation of 789 s. With the measured hopping speed, only more than +7 cases would lead to missing activated hops.

Partial Sequence Extraction. Following the basic analysis of the target system, we extracted the partial hopping sequences. As described in the previous subsection, the total bandwidth of the target system was approximately 70

MHz, much higher than the maximum bandwidth of the USRP (25 MHz). Therefore, we first acquired the partial hopping sequences of the target system.

In order to record the sequence, we built a GNU Radio flow graph as in Fig. 4. The flow graph was mainly composed of a Frequency Xlating FIR filter, Power Squelch, and Function Probe.

```

build flow graph structure;
while running flow graph do
  for each interval do
    for each channel do
      if squelch active then
        record(channel_number);
      end
      if no activation then
        record(0);
      end
    end
  end
end
end

```

Algorithm 1: Partial sequence extraction algorithm

Frequency Xlating FIR filter first operates as a channel selection filter. It tunes to the target channel and filters out other signals. Power Squelch and Function Probe are core parts of this flow graph. Power Squelch allows input signals to pass through only when the power level exceeds a preset threshold, and Function Probe monitors the state of Power Squelch to determine if it is open.

We parallelized the flow graph in Fig. 4 to simultaneously record multiple channels under USRP coverage. We set seven USRP coverages, and ran Algorithm 1 for each coverage to record the corresponding partial sequence. As a result, we finally acquired all partial hopping sequences, as listed in Tab. 1. From the table, we see that all channels were identically activated three times for each partial period. This confirmed that the target system uniformly unitizes its bandwidth.

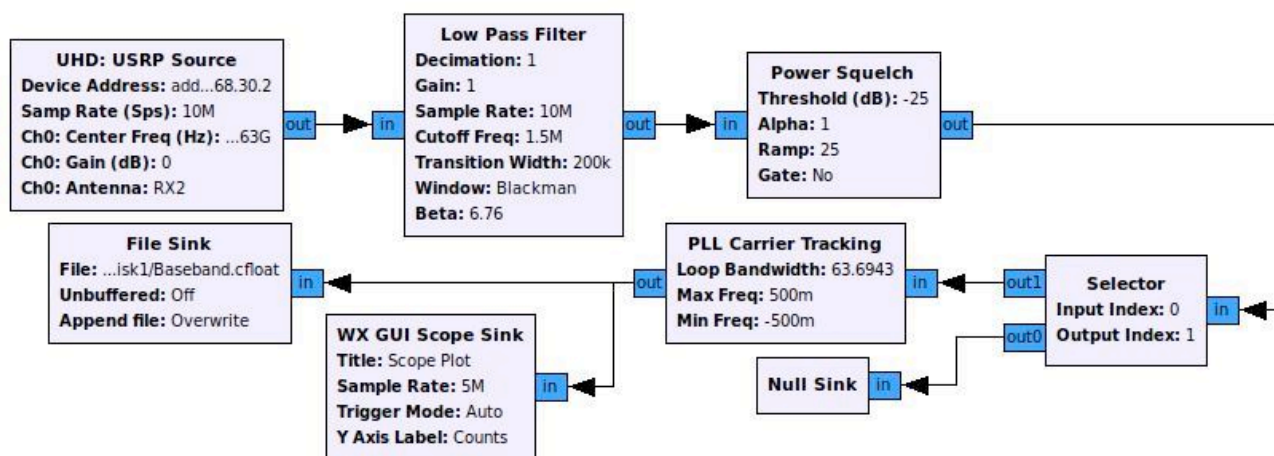


Fig. 5: GNU radio flow graph for the baseband extractor

Combining Partial Sequences. In the final step, we combined acquired partial sequences. This step was not automated, since partial sequences can be woven manually due to their short lengths. We arranged partial sequences in a spreadsheet, and fit the activations of the overlapped channels together to find the complete hopping sequence.

Though the coverages did not overlap maximally, a unique combination could be found. Tab. 2 shows the combined sequence.

Table 1: Extracted partial sequences for each coverage

Coverage #	Partial hopping sequence	Length
1 (Ch1 Ch9)	7,1,6,5,4,9,3,8,2,7,1,6,5,4,3,2,1,9,8,7,6,5,4,9,3,8,2	27
2 (Ch1 Ch17)	7, 1, 12, 6, 11, 5, 10, 4, 9, 3, 8, 2, 7, 1, 6, 17, 5, 16, 4, 15, 3, 14, 2, 13, 1, 12, 17, 11, 16, 10, 15, 9, 14, 8, 13, 7, 12, 6, 17, 11, 5, 16, 10, 4, 15, 9, 3, 14, 8, 2, 13	51
3 (Ch9 Ch25)	12, 23, 11, 22, 10, 21, 9, 20, 25, 19, 24, 18, 23, 17, 22, 16, 21, 15, 20, 14, 25, 19, 13, 24, 18, 12, 23, 17, 11, 22, 16, 10, 21, 15, 9, 20, 14, 19, 13, 18, 12, 17, 11, 16, 10, 15, 9, 14, 25, 13, 24	51
4 (Ch17 Ch33)	26, 31, 25, 30, 24, 29, 23, 28, 22, 33, 27, 21, 32, 26, 20, 31, 25, 19, 30, 24, 18, 29, 23, 17, 28, 22, 27, 21, 26, 20, 25, 19, 24, 18, 23, 17, 22, 33, 21, 32, 20, 31, 19, 30, 18, 29, 17, 28, 33, 27, 32	51
5 (Ch25 Ch41)	41, 29, 40, 28, 39, 27, 38, 26, 37, 25, 36, 41, 35, 40, 34, 39, 33, 38, 32, 37, 31, 36, 30, 41, 35, 29, 40, 34, 28, 39, 33, 27, 38, 32, 26, 37, 31, 25, 36, 30, 35, 29, 34, 28, 33, 27, 32, 26, 31, 25, 30	51
6 (Ch33 Ch47)	44, 43, 42, 47, 41, 46, 40, 45, 39, 44, 38, 43, 37, 42, 36, 47, 41, 35, 46, 40, 34, 45, 39, 33, 44, 38, 43, 37, 42, 36, 41, 35, 40, 34, 39, 33, 38, 37, 36, 47, 35, 46, 34, 45, 33	45
7 (Ch39 Ch47)	44, 43, 42, 47, 41, 46, 40, 45, 39, 44, 43, 42, 47, 41, 46, 40, 45, 39, 44, 43, 42, 41, 40, 39, 47, 46, 45	27

Table 2: Acquired total hopping sequence

Combined partial periods	7, 1, 36, 30, 24, 12, 6, 47, 35, 29, 23, 11, 5, 46, 34, 28, 22, 10, 4, 45, 33, 27, 21, 9, 3, 44, 32, 26, 20, 8, 2, 43, 31, 25, 19, 7, 1, 42, 30, 24, 18, 6, 47, 41, 29, 23, 17, 5, 46, 40, 28, 22, 16, 4, 45, 39, 27, 21, 15, 3, 44, 38, 26, 20, 14, 2, 43, 37, 25, 19, 13, 1, 42, 36, 24, 18, 12, 47, 41, 35, 23, 17, 11, 46, 40, 34, 22, 16, 10, 45, 39, 33, 21, 15, 9, 44, 38, 32, 20, 14, 8, 43, 37, 31, 19, 13, 7, 42, 36, 30, 18, 12, 6, 41, 35, 29, 17, 11, 5, 40, 34, 28, 16, 10, 4, 39, 33, 27, 15, 9, 3, 38, 32, 26, 14, 8, 2, 37, 31, 25, 13 (Length = $47 \times 3 = 141$)
--------------------------	--

4.4 Baseband Extractor

Having successfully extracted the hopping sequence, we programmed the USRP to follow and record the target FHSS signal. In order to do that, we built a GNU radio flow graph as in Fig. 5. In the flow graph, the incoming signal is first filtered by Low Pass Filter block. Power Squelch then senses the activeness of the current channel. Selector is initially headed to the null sink in order not to record meaningless signals, and is switched to the remaining flow graph when the USRP catches the FHSS signal. Once Selector is switched, PLL Carrier Tracking block finely tunes on the signal stream. Finally, the signal stream is recorded to an output file and visualized in real time. The flow graph is

dynamically controlled using a Python script. It first commands the USRP to monitor one of the channels. When the channel being awaited is activated, the script compares the state of its internal counter at the moment with the incoming signal. If they match, the script switches Selector and starts recording the signal stream. Otherwise, it is reset. As a result, we successfully extracted the raw baseband signal of the target system. Fig. 6 shows a part of the extracted signal.

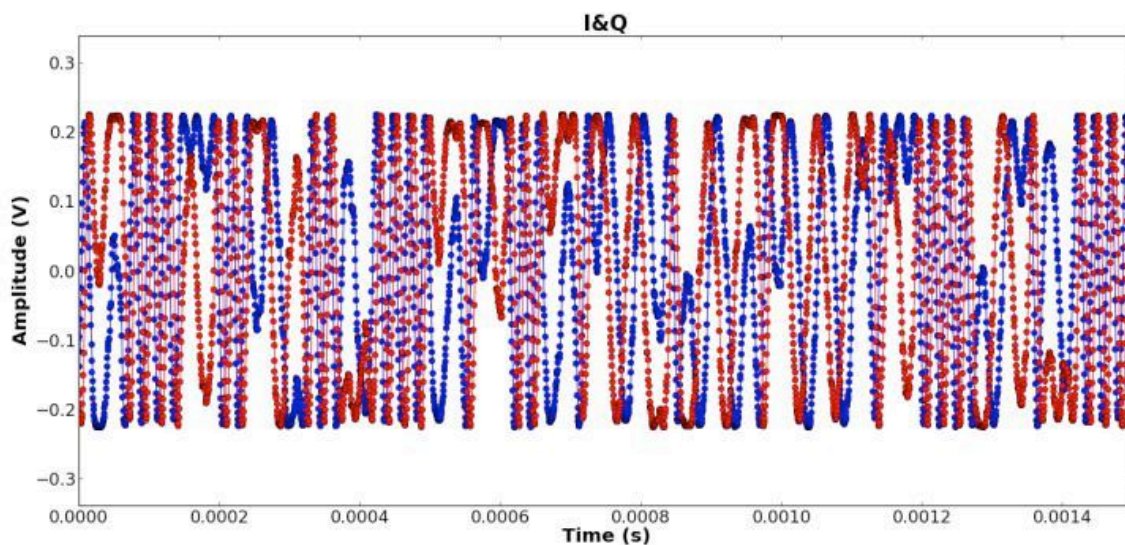


Fig. 6: Part of the extracted baseband

5. Related Work

5.1 Drone Security

Several attack trials have shown that drones are quite vulnerable. With regard, for instance, to the RQ-170 USAF drone mentioned in the introduction, the Iranian government claimed that it had captured the drone through its cyber-warfare unit [21]. Although some debates on the attack means exist, the captured drone seems quite intact, which means it was not shot down by projectiles.

With regard to civilian drones, Todd Humphreys et al. insisted that civilian drones are threatened by GPS spoofing [10]. It was even demonstrated that drones are hijacked by spoofing GPS signals with a custom GPS spoofer [10]. Kamkar recently announced SkyJack, a specialized drone hijack platform that targets only Parrot AR drones [12]. It exploits a WiFi hotspot vulnerability in AR drones to acquire control over them.

5.2 FHSS Security

FHSS is widely adopted to various communication devices for the motive of securing transmission, as its rapid pseudorandom frequency shift apparently makes FHSS systems resilient against eavesdroppers or jammers to a certain extent. However, much research indicates FHSS alone cannot completely secure the contents being transferred. Song et al. presented several algorithms for breaking the pseudorandom FHSS sequence with external observation [16]. Presented algorithms were theoretically analyzed, and some were stimulated with C++ software,

which is different from our work where the attack scheme is implemented and verified in reality. Furthermore, Song et al. assume omniscience in receiving the target signal, and thus limited receiver bandwidth was not considered in their work. Q et al. utilized a low-cost hardware equipped with commercial RF Integrated Circuits (RFIC) to implement a hopping sequence analyzer [14]. With the tool implemented, they successfully extracted hopping patterns in the 902-928MHz spectrum. However, their work is different from ours in some aspects. First, the presented approach can only be applied to a spectrum to which the RFIC used can be tuned, whereas our SDR approach is much more flexible. Second, overcoming the limited receiver bandwidth was not covered in their work.

5.3 Bluetooth Security

Bluetooth is among the best-known communication standards that use FHSS. The wide adoption of Bluetooth in input devices suggests the likelihood of critical attacks. Bluetooth Low Energy (BLE) especially adopts a much simpler hopping and key sharing mechanism than classic Bluetooth. Mike Ryan has claimed [15] that the hopping sequence of BLE can be identified by collecting empty data packets, and attackers can sniff ongoing links. He used Ubertooth, a programmable BLE sniffer, to extract parameters required to acquire the hopping sequence, and brute-forced the encryption key, which enabled BLE sniffing.

6. Discussion and Future Works

Attack research on drones has not only the meaning of attacking a system. It is also highly related to privacy protection, infrastructure security, and defense, since drones are becoming severe threats against them. A drone control system is apparently one of the major attack vectors against drones. This work deals with the very first step of attacking an FHSS-type control system by acquiring the hopping sequence, which is essential to realize attacks.

For future works, first, we will analyze the baseband signal to reveal its structure. If it is not encrypted, our attack platform can operate as a sniffer, which can monitor control signals. This will enable the attacker to predict the movement of the target drone. Additionally, carefully crafted spoofing waveforms can take control of the target drone, which will give the defender a way to safely capture the target drone. Second, we will automate the process of combining partial hopping sequences to make the presented attack scheme applicable to general wideband FHSS devices.

7. Conclusion

In this article we proposed a general scheme to extract the hopping sequences of FHSS-type RC drone controllers and showed its effectiveness using an SDR. We also proposed a scheme to overcome the issue of the limited bandwidth of the SDR and showed that it was effective by successfully extracting the baseband signal of a target system in an experiment. Our work can be extended to be implemented on jammers, sniffers, and spoofers against RC controllers.

References:

1. 2.4GHz Radio Control Explained,
<http://www.rcmodelreviews.com/spreadspectrum01.shtml>
2. Amazon Prime Air, <http://www.amazon.com/b?node=8037720011>
3. CBX 1200-6000 MHz Rx/Tx (40 MHz), <http://www.ettus.com/product/details/CBX>
4. DHL launches first commercial drone ‘parcelcopter’ delivery service,
<http://www.theguardian.com/technology/2014/sep/25/german-dhl-launches-first-commercial-drone-delivery-service>
5. How compatible are 2.4GHz RC systems?,
<http://www.rcmodelreviews.com/rxcompatibility.shtml>
6. USRP N210 Datasheet,
www.ettus.com/content/files/~07495_Ettus_N200-210_DS_Flyer_HR_1.pdf
7. Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., Goldbaum, M.: Detection of blood vessels in retinal images using two-dimensional matched filters. IEEE T-MI 8(3), 263{269 (1989)
8. Chen, Q., Defrise, M., Deconinck, F.: Symmetric phase-only matched filtering of fourier-mellin transforms for image registration and recognition. TPAMI 16(12), 1156{1168 (1994)
9. Gardner, W., et al.: Exploitation of spectral redundancy in cyclostationary signals. IEEE Signal Processing Magazine 8(2), 14{36 (1991)
10. Humphreys, T.E., Ledvina, B.M., Psiaki, M.L., O'Hanlon, B.W., Kintner Jr, P.M.: Assessing the spoofing threat: Development of a portable gps civilian spoofer. In: ION GNSS+. vol. 55, p. 56 (2008)
11. James, M.: What are DSM RC Controllers and Receivers and What Do They Do?,
<http://rcvehicles.about.com/od/frequency/f/dsmtechnology.htm>
12. Kamkar, S.: SkyJack, <http://www.samy.pl/skyjack/>
13. Nychis, G., Hottelier, T., Yang, Z., Seshan, S., Steenkiste, P.: Enabling macprotocol implementations on software-defined radios. In: NSDI. vol. 9, pp. 91{105 (2009)
14. Q, Atlas, Cutaway Smash, Slugs on Toast: Hop hacking hedy (2011),
<https://www.youtube.com/watch?v=aMBaO94Q49U>

References:

15. Ryan, M.: Bluetooth: With low energy comes low security. In: WOOT (2013)
16. Song, M., Allison, T.: Frequency hopping pattern recognition algorithms for wireless sensor networks. In: ISCA. pp. 264{269 (2005)
17. Stahlberg, M.: Radio jamming attacks against two popular mobile networks. In: Tik-110.501. vol. 3 (2000)
18. Tang, H.: Some physical layer issues of wide-band cognitive radio systems. In: DySPAN. pp. 151{159. IEEE (2005)
19. Urkowitz, H.: Energy detection of unknown deterministic signals. Proceedings of the IEEE 55(4), 523{531 (1967)
20. Wikipedia: General Atomics MQ-1 Predator (2015), http://en.wikipedia.org/wiki/General_Atomics_MQ-1_Predator
21. Wikipedia: Iran-U.S. RQ-170 incident (2015), http://en.wikipedia.org/wiki/Iran%E2%80%93U.S._RQ-170_incident
22. Wikipedia: Sequence alignment (2015), http://en.wikipedia.org/wiki/Sequence_alignment
23. Yucek, T., Arslan, H.: Spectrum characterization for opportunistic cognitive radio systems. In: MILCOM. pp. 1{6. IEEE (2006)
24. Yucek, T., Arslan, H.: A survey of spectrum sensing algorithms for cognitive radio applications. IEEE Communications Surveys & Tutorials 11(1), 116{130 (2009)



Wireless Communications with UAV: Opportunities and Challenges

Yong Zeng

Rui Zhang

Teng Joon Lim



ABOUT THE AUTHOR

YONG ZENG

Yong Zeng received the B.E. (first-class honors) and Ph.D. degrees from Nanyang Technological University, Singapore, in 2009 and 2014, respectively. From 2013 to 2018, he was a Research Fellow and a Senior Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore. From 2018 to 2019, he was a Lecturer with the School of Electrical and Information Engineering, The University of Sydney. He is currently with the National Mobile Communications Research Laboratory, Southeast University, and the Purple Mountain Laboratories, Nanjing, China.



ABOUT THE AUTHOR

RUI ZHANG

I got my Ph.D. degree from the Department of Electrical Engineering of Stanford University in 2007. During my doctoral studies at Stanford, I was with the Dynamic Spectrum Management Laboratory under the supervision of Prof. John M. Cioffi. From 2007 to 2009, I worked at the Institute for Infocomm Research in Singapore as a researcher. Since January 2010, I have joined the Department of Electrical and Computer Engineering of National University of Singapore, where I am now a Professor.



ABOUT THE AUTHOR

TENG JOON LIM

My research interests have evolved over the last 20 years from adaptive signal processing in my PhD years, to multi-user detection and CDMA transceiver design in my CWC years, to MIMO processing, cooperative diversity, phase, frequency and channel estimation in multi-carrier modulation, spectrum sensing for cognitive radio, and wireless resource allocation in my U of T years. Since joining NUS in mid-2011, I have worked on green communications, smart grid and heterogeneous networks, satellite communications, and network architecture and cyber-security in the Internet of Things.

Wireless communication systems that include unmanned aerial vehicles (UAVs) promise to provide cost-effective wireless connectivity for devices without infrastructure coverage. Compared to terrestrial communications or those based on high-altitude platforms (HAPs), on-demand wireless systems with low-altitude UAVs are in general faster to deploy, more flexibly re-configured, and are likely to have better communication channels due to the presence of short-range line-of-sight (LoS) links. However, the utilization of highly mobile and energy-constrained UAVs for wireless communications also introduces many new challenges. In this article, we provide an overview of UAV-aided wireless communications, by introducing the basic networking architecture and main channel characteristics, highlighting the key design considerations as well as the new opportunities to be exploited.

1. INTRODUCTION

With their high mobility and low cost, unmanned aerial vehicles (UAVs), also commonly known as drones or remotely piloted aircrafts, have found a wide range of applications during the past few decades [1]. Historically, UAVs have been primarily used in the military, mainly deployed in hostile territory to reduce pilot losses. With the continuous cost reduction and device miniaturization, small UAVs (typically with weight not exceeding 25 kg) are now more easily accessible to the public and thus numerous new applications in civilian and commercial domains have emerged, with typical examples including weather monitoring, forest fire detection, traffic control, cargo transport, emergency search and rescue, communication relaying, etc [2]. UAVs can be broadly classified into two categories: fixed wing versus rotary wing, each with their own strengths and weaknesses. For example, fixed-wing UAVs usually have high speed and heavy payload, but they must maintain a continuous forward motion to remain aloft, thus are not suitable for stationary applications like close inspection. In contrast, rotary-wing UAVs such as quadcopters, though having limited mobility and payload, are able to move in any direction as well as to stay stationary in the air. Thus, the choice of UAVs critically depends on the applications.

Among the various applications enabled by UASs, the use of UAVs for achieving high-speed wireless communications is expected to play an important role in future communication systems. In fact, UAV-aided wireless communication offers one promising solution to provide wireless connectivity for devices without infrastructure coverage due to severe shadowing by urban or mountainous terrain, for example, or damage to the communication infrastructure caused by natural disasters [3]. Note that besides UAVs, one alternative solution for wireless connectivity is via high-altitude platforms (HAPs), such as balloons, which usually operate in the stratosphere that is tens of kilometers above the Earth's surface. HAP-based communications have several advantages over the UAV-based low-altitude platforms (LAPs), such as wider coverage, longer endurance, etc. Thus, HAP is in general preferred for providing reliable wireless coverage for a large geographic area. However, compared to HAP-based communications, or those based on terrestrial or satellite systems, wireless communications with low-altitude UAVs (typically at an altitude not exceeding several kilometers) also have several important advantages. First, on-demand UASs are more cost-effective and can be much more swiftly deployed, which makes them especially suitable for unexpected or limited-duration missions. Besides, with the aid of low-altitude UAVs, short-range line-of-sight (LoS) communication links can be established in most scenarios, which potentially leads to significant performance improvement over direct communication between

source and destination (if possible) or HAP relaying over long-distance LoS links. In addition, the maneuverability of UAVs offers new opportunities for performance enhancement, through the dynamic adjustment of UAV state to best suit the communication environment. Furthermore, adaptive communications can be jointly designed with UAV mobility control to further improve the communication performance. For example, when a UAV experiences good channels with the ground terminals, besides transmitting with higher rates, it can also lower its speed to sustain the good wireless connectivity to transmit more data to the ground terminals. These evident benefits make UAV-aided wireless communication a promising integral component of future wireless systems, which need to support more diverse applications with orders-of-magnitude capacity improvement over the current systems. Fig. 1 illustrates three typical use cases of UAV-aided wireless communications, which are:

UAV-aided ubiquitous coverage, where UAVs are deployed to assist the existing communication infrastructure, if any, in providing seamless wireless coverage within the serving area. Two example scenarios are rapid service recovery after partial or complete infrastructure damage due to natural disasters, and base station offloading in extremely crowded areas, e.g., a stadium in a sports event. Note that the latter case has been identified as one of the five key scenarios that need to be effectively addressed by the fifth generation (5G) wireless systems [4].

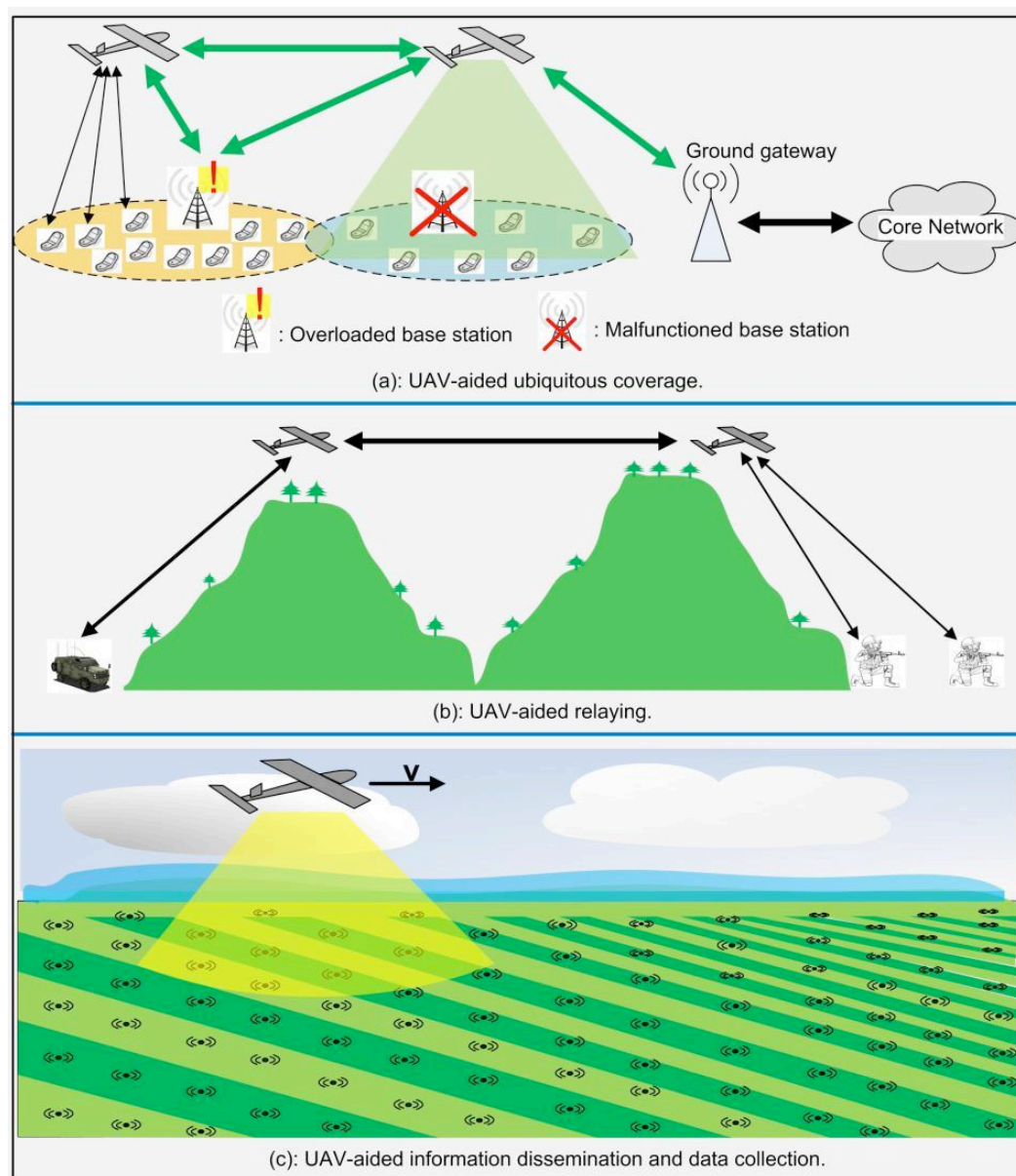


Fig. 1: Three typical use cases of UAV-aided wireless communications.

UAV-aided relaying, where UAVs are deployed to provide wireless connectivity between two or more distant users or user groups without reliable direct communication links, e.g., between the frontline and the command center for emergency responses.

UAV-aided information dissemination and data collection, where UAVs are dispatched to disseminate (or collect) delay-tolerant information to (from) a large number of distributed wireless devices, e.g., wireless sensors in precision agriculture applications.

Despite the many promising benefits, wireless communications with UAVs are also faced with several new design challenges. First, besides the normal communication links as in terrestrial systems, additional control and non-payload communications (CNPC) links with much more stringent latency and security requirements are needed in UASs for supporting safety-critical functions, such as real-time control, collision and crash avoidance, etc. This calls for more effective resource management and security mechanisms specifically designed for UAV communication systems. Besides, the high mobility environment of UASs generally results in highly dynamic network topologies, which are usually sparsely and intermittently connected [5]. As a result, effective multi-UAV coordination, or UAV swarm operations, need to be designed for ensuring reliable network connectivity [6]. At the same time, new communication protocols need to be designed taking into account the possibility of sparse and intermittent network connectivity. Another main challenge stems from the size, weight, and power (SWAP) constraints of UAVs, which could limit their communication, computation, and endurance capabilities. To tackle such issues, energy-aware UAV deployment and operation mechanisms are needed for intelligent energy usage and replenishment. Moreover, due to the mobility of UAVs as well as the lack of fixed backhaul links and centralized control, interference coordination among the neighboring cells with UAV-enabled aerial base stations is more challenging than in terrestrial cellular systems. Thus, effective interference management techniques specifically designed for UAV-aided cellular coverage are needed.

The objective of this article is to give an overview of UAV-aided wireless communications. The basic networking architecture, main channel characteristics and design considerations, as well as the key performance enhancing techniques that exploit the UAV's mobility, will be presented.

2. BASIC NETWORKING ARCHITECTURE

Fig. 2 shows the generic networking architecture of wireless communications with UAVs, which consists of two basic types of communication links, namely the CNPC link and the data link.

A. Control and Non-Payload Communications Link

The CNPC links are essential to ensure the safe operation of all UASs. Highly reliable, low-latency, and secure two-way communications, usually with low data rate requirement, must be supported by these links for exchanging safety-critical information among UAVs, as well as between the UAV and ground control stations (GCS), e.g.,

dedicated mobile terminals mounted on ground vehicles. The main CNPC information flow can be broadly categorized into three types: i) command and control from GCS to UAVs; aircraft status report from UAVs to ground; iii) sense-and-avoid information among UAVs. Even for autonomous UAVs, which are able to accomplish missions relying on onboard computers without real-time human control, the CNPC links are also necessary in case emergency human intervention is needed. Not shown in Fig. 2 are the air traffic control (ATC) links, which are necessary only when the UAVs are within a controlled airspace, e.g., near an airport.

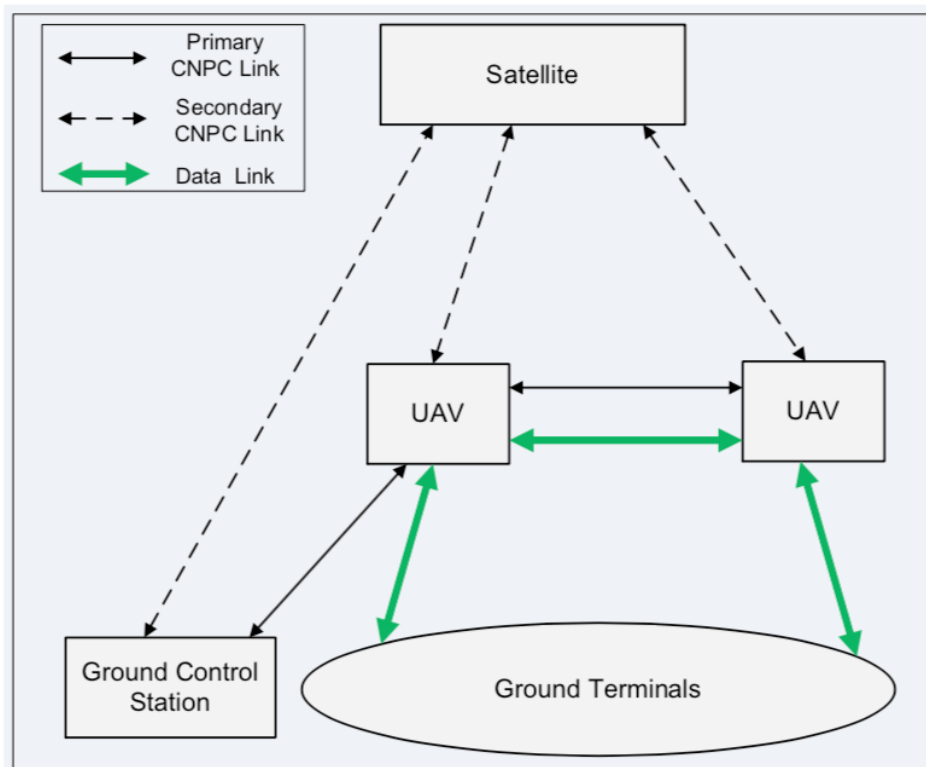


Fig. 2: Basic networking architecture of UAV-aided wireless communications.

Due to the critical functions to be supported, CNPC links should in general operate in protected spectrum. Currently two such bands have been allocated, namely the L-band (960-977MHz) and the C-band (5030-5091MHz) [7]. Furthermore, although the direct links between GCS and UAVs (primary CNPC links) are always preferred for delay reasons, secondary CNPC links via satellite could also be exploited as a backup to enhance reliability and robustness. Another key requirement for CNPC links is the superior high security. In particular, effective security mechanisms should be employed to avoid the so-called ghost control scenario, a potentially catastrophic situation in which the UAVs are controlled by unauthorized agents via spoofed control or navigation signals. Therefore, powerful authentication techniques, possibly complemented by the emerging physical layer security techniques, should be applied for CNPC links.

B. Data Link

The data links, on the other hand, aim to support mission-related communications for the ground terminals, which, depending on the application scenarios, may include terrestrial base stations (BSs), mobile terminals, gateway nodes, wireless sensors, etc. Taking the UAV-aided ubiquitous coverage shown in Fig 1(a) as an example, the data links maintained by the UAVs need to support the following communication modes: i) direct mobile-UAV communication as for BS offloading or during complete BS malfunction; ii) UAV-BS and UAV-gateway wireless backhaul; iii)

UAV-UAV wireless backhaul. The capacity requirement for these data links critically depends on the applications, possibly ranging from several kbps in UAV-sensor links to dozens of Gbps in UAV-gateway wireless backhaul. Compared to CNPC links, the data links usually have higher tolerance in terms of latency and security requirements. In terms of spectrum, the UAV data links could reuse the existing band assigned for the particular applications to be supported, e.g., the LTE band while assisting cellular coverage, or a dedicated new spectrum could be allocated for enhanced performance, e.g., using millimeter wave (mmWave) band for high capacity UAV-UAV wireless backhaul [8].

3. CHANNEL CHARACTERISTICS

Both CNPC and data links in UAV-aided communications consist of two types of channels, namely UAV-ground and UAV-UAV channels, which exhibit several unique characteristics as compared to the extensively studied terrestrial communication channels.

A. UAV-Ground Channel

While the air-ground channels for aeronautical applications with piloted aircrafts are well understood, systematic measurements and modeling of UAV-ground channels are still ongoing [7], [9]. Unlike piloted aircraft systems, where the ground sites are usually in open areas with tall antenna towers, the UAV-ground channels for UASs are more complicated due to the more complex operational environment. While LoS links are expected for such channels in most scenarios, they could also be occasionally blocked by obstacles such as terrain, buildings, or the airframe itself. In particular, recent measurements have shown that the UAV-ground channels could suffer from severe airframe shadowing with a duration up to dozens of seconds during aircraft maneuvering [9], which needs to be taken into account for mission-critical operations. For low-altitude UAVs, the UAV-ground channels may also constitute a number of multi-path components due to reflection, scattering, and diffraction by mountains, ground surface, foliage, etc. For UAVs operating over desert or sea, the two-ray model has been mostly used due to the dominance of the LoS and the surface reflection components. Another widely used model is the stochastic Rician fading model, which consists of a deterministic LoS component, and a random scattered component with certain statistical distributions. Depending on the environment surrounding the ground terminals as well as the frequency used, the UAV-ground channels exhibit widely varying Rician factors, i.e., the power ratio between the LoS and the scattered components, with typical values around 15 dB for L-band and 28 dB for C-band in hilly terrain [7].

B. UAV-UAV Channel

The UAV-UAV channels are mainly dominated by the LoS component. Although there may exist limited multipath fading due to ground reflections, its impact is minimal as compared to that experienced in UAV-ground or ground-ground channels. In addition, the UAV-UAV channels may have even higher Doppler frequencies than the UAV-ground counterparts, due to the potentially large relative velocity between UAVs. Such channel characteristics have direct implications on spectrum allocation for UAV-UAV links. On one hand, the dominance of LoS links may

suggest that the emerging mmWave communications could be employed to achieve high-capacity UAV-UAV wireless backhaul. On the other hand, the high relative velocity between UAVs coupled with the higher frequency in the mmWave band could lead to excessive Doppler shift. More in-depth studies are needed to find out the most suitable technology to use in UAV-UAV links, given their unique channel characteristics.

4. MAIN DESIGN CONSIDERATIONS

This section presents the main design considerations specifically for wireless communications with UAVs. The following three aspects are discussed: UAV path planning, energy-aware deployment and operation, and multiple-input multiple-output (MIMO) communications in UASs.

A. UAV Deployment and Path Planning

One important design aspect of UASs is UAV path planning [10], [11]. For UAV-aided communications in particular, appropriate path planning may significantly shorten the communication distance and thus is crucial for high-capacity performance. Unfortunately, finding the optimal flying path for UAV is a challenging task in general. On one hand, UAV path optimization problems essentially involve an infinite number of variables due to the continuous UAV trajectory to be determined. On the other hand, the problems are also usually subject to a variety of practical constraints, e.g., connectivity, fuel limitation, collision and terrain avoidance, many of which are time-varying in nature and are difficult to model accurately. One useful method for UAV path planning is to approximate the UAV dynamics by a discrete-time state space, with the state vector typically consisting of the position and velocity in a three-dimensional (3D) coordinate system. The UAV trajectory is then given by the sequence of states, which are subject to finite transition constraints to reflect the practical UAV mobility limitations. Many of the resulting problems with such an approximation belong to the class of mixed integer linear programming (MILP) [11], which can be solved with well-developed software packages.

Intuitively, the optimal UAV flight path critically depends on the application scenarios. For instance, for UAV-aided cellular coverage as shown in Fig. 1(a), it is evident that more than one UAV should be jointly deployed above the serving areas to cooperatively achieve real-time communications with ground users; whereas for UAV-aided information dissemination or collection for delay-tolerant data, as shown in Fig. 1(c), it could be sufficient to dispatch one single UAV to fly over the area to communicate with the ground nodes sequentially. Furthermore, for the cellular coverage application, one option is to employ rotary-wing UAVs that hover above the coverage area, serving as static aerial base stations. In this case, no dedicated path planning is needed. Instead, the main design problems for UAV deployment usually involve finding the optimal UAV separations as well as their hovering altitude to achieve maximum coverage. Note that for a typical urban environment, there in general exists an optimal UAV altitude in terms of coverage maximization, which is due to the following non-trivial tradeoff: While increasing UAV altitude will lead to higher free space path loss, it also increases the possibility of having LoS links with the ground terminals. Such a tradeoff has been characterized in [12], [13], based on which the optimal UAV altitude has been obtained.

B. Energy-Aware Deployment and Operation

The performance and operational duration of a UAS is fundamentally constrained by the limited onboard energy. Although powerplant and energy-storage technologies have advanced dramatically over the past few decades, limited energy availability still severely hampers UAV endurance. From the operational perspective, this problem can be addressed through two approaches. First, effective energy-aware deployment mechanisms are needed for timely onboard energy replenishment, yet without noticeable interruption of the communication services supported. Second, energy-efficient operation through smart energy management is required, i.e., accomplishing the missions with minimum energy consumption.

In terms of energy-aware deployment, one effective approach is to exploit the inter-UAV cooperation to enable sequential energy replenishment. For instance, at any one time, only one UAV is scheduled to leave the serving area for energy replenishment, during which the service gap is temporarily filled by neighboring UAVs via e.g., increasing the transmission power and/or adjusting the aircraft positions. This energy replenishment scheduling can be matched to the dynamic load patterns that need to be supported by the UAVs. For instance, it might be preferred to schedule energy replenishment only when low data traffic is expected, e.g., during night time for the cellular coverage application. Note that apart from the commonly used energy sources, such as electric batteries or liquid fuels, there has been increasing interest in powering UAVs by solar energy or dedicated wireless energy transfer technology via laser beams, for example.

Energy-efficient operation, on the other hand, aims to reduce unnecessary energy consumption by the UAVs. As the main energy usage of UAVs is to support either aircraft propulsion or wireless communications, energy-efficient operation schemes can be broadly classified into two categories. The first one is energy-efficient mobility, for which the movement of the UAVs should be carefully controlled by taking into account the energy consumption associated with every maneuver. For instance, unnecessary aircraft maneuvering or ascending should be avoided since they are generally quite energy-intensive. Energy-efficient mobility schemes can usually be designed with path planning optimization, by using appropriate energy consumption models as a function of UAV speed, acceleration, altitude, etc. The other category of energy-efficient operation is energy-efficient communication, which aims to satisfy the communication requirement with the minimum energy expenditure on communication-related functions, such as communication circuits, signal transmission, etc. To this end, one common approach is to optimize the communication strategies to maximize the energy efficiency (EE) in bits/Joule, i.e., the number of successfully communicated data bits per unit energy consumption. Note that while energy-efficient communication has been extensively studied for terrestrial communications, its systematic investigation for UAV communication systems is still under-developed.

C. MIMO for UAV-Aided Communications

Although MIMO technology has been extensively implemented in terrestrial communication systems due to its high spectral efficiency and superior diversity performance, its application in UASs is still hindered by several factors. First, the lack of rich scattering in the UAS environment considerably limits the spatial multiplexing gain of MIMO, which

usually leads to only marginal rate improvement over single-antenna systems. Besides, the high signal processing complexity as well as the hardware and power consumption costs make it quite costly to employ multiple antennas in UAVs due to the SWAP limitations. Furthermore, MIMO systems rely on accurate channel state information (CSI) for best performance. However, this is practically difficult to achieve in a highly dynamic environment, therefore further limiting the practical MIMO gain in UASs.

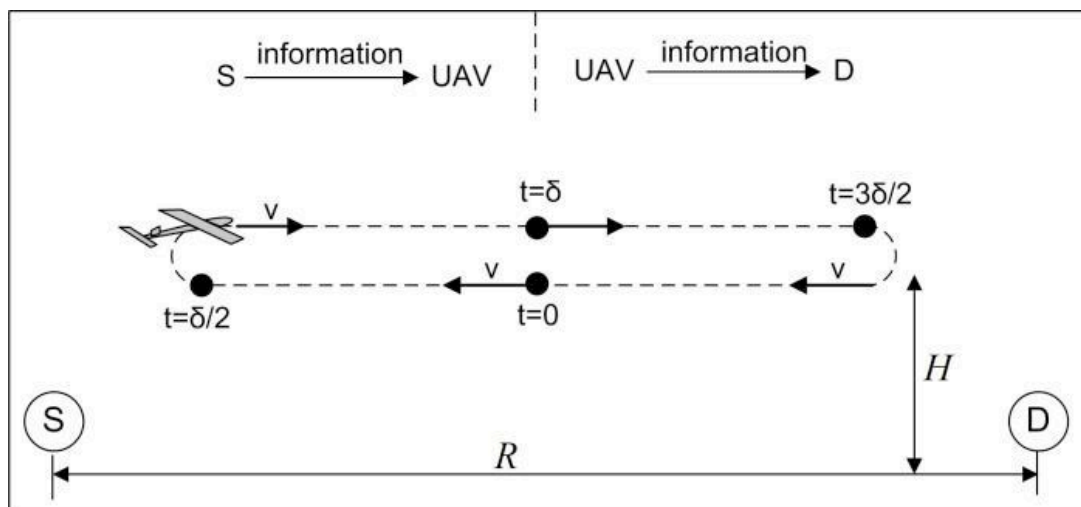
Despite the above challenges, some recent results still show a great potential for MIMO technology in UASs. In particular, in contrast to the common conception that spatial multiplexing gain is fundamentally limited by the number of signal paths, it has been found that high spatial multiplexing gain may also be attainable even in LoS channels, by carefully designing the antenna separation with respect to carrier wavelength and link distance [14], though this usually requires large antenna separation, high carrier frequency, and short communication range. Alternatively, a more practical way to reap the multiplexing gain in a poor scattering environment is to leverage multi-user MIMO, by simultaneously serving a number of sufficiently separated ground terminals with angular separations exceeding the angular resolution of the antenna array installed on the UAVs. In this case, the signals for different terminals are distinguishable by the UAV array, and thus restores the MIMO spatial multiplexing gain. Another way of utilizing MIMO in UASs is through mmWave communications, for which the MIMO array gain, instead of the spatial multiplexing gain, is more critical due to the large available bandwidth as well as the high signal attenuation. However, due to the high mobility of UAVs, it would be quite challenging to achieve transmitter/receiver beam alignment for directional mmWave communications, an issue that needs to be properly addressed before mmWave MIMO could be practically employed in UAV systems.

5. COMMUNICATIONS WITH UAV CONTROLLED MOBILITY

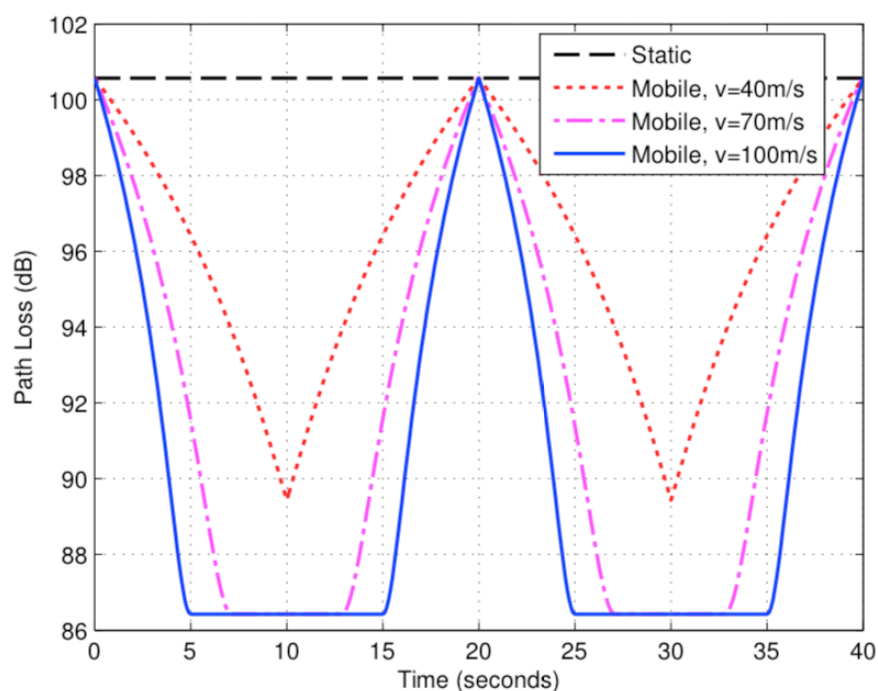
The high mobility of UAVs offers unique opportunities for performance improvement in UAV-aided communications. In this section, we discuss two key techniques for wireless communications with UAV controlled mobility, which are UAV-enabled mobile relaying and device-to-device (D2D)-enhanced UAV information dissemination.

A. UAV-Enabled Mobile Relaying

Relaying is an extensively studied technique in terrestrial communication systems for throughput/reliability improvement as well as range extension. Due to the practical constraints such as limited mobility and wired backhauls, most relays in terrestrial systems are deployed in fixed locations, which we term as static relaying. To further exploit the UAV controlled mobility, we present in this subsection a UAV-enabled mobile relaying strategy, which works particularly well for delay-tolerant applications.



(a) A schematic of the UAV-enabled mobile relaying



(b) Path loss with static versus mobile relaying

Fig. 3: UAV-enabled mobile relaying and the corresponding path loss of the communication links.

With mobile relaying, the UAV flies continuously between the source and destination aiming to reduce the link distances during both UAV information reception and relaying phases. For example, with half-duplex decode-and-forward (DF) mobile relaying, each relaying cycle consists of two phases each with duration δ seconds, where δ is determined by the maximum tolerable delay. As illustrated in Fig. 3(a), the first phase corresponds to UAV information reception, where it keeps receiving and decoding the information sent from the source and stores in its data buffer. Concurrently, starting from the initial position at the middle point between the source and destination, the UAV first flies towards the source at a maximum possible speed v , and then flies back timely so that it returns to the initial position at the end of the first phase ($t = \delta$). Note that if v and/or δ is sufficiently large, the UAV will have time to hover above the source before returning so as to enjoy the best channel for data reception. In the second phase starting from $t = \delta$, the UAV sends the data in its buffer to the destination. This is accompanied by a symmetric UAV movement, where it first flies towards the destination, hovers above the nearest location to the destination if time allows, and then returns to the initial position at the end of the cycle ($t = 2\delta$). It is evident that compared to static relaying with the fixed UAV location at the same initial position, the proposed mobile relaying strategy always enjoys a

shorter link distance (or better average channel) in each of the two phases of information reception and relaying. This is illustrated in Fig. 3(b) with $\delta = 20$ seconds under different UAV velocity and a constant height $H = 100$ m. The carrier frequency is 5 GHz and the source and destination are assumed to be separated by $R = 1$ km. It is observed from Fig. 3(b) that with higher UAV speed limitation, mobile relaying enjoys larger link gains over static relaying. In particular, for sufficiently large UAV maximum speed, e.g., $v = 100$ m/s, the UAV would be able to stay stationary above the source and destination each for about 10 seconds, during which the path loss remains at a constant value that is about 14 dB lower than that of the static relaying.

By employing adaptive rate transmission, the proposed mobile relaying strategy can achieve significant throughput improvement over the conventional static relaying. This is illustrated in Fig. 4, where the end-to-end spectrum efficiency in bps/Hz is plotted against the maximum tolerable delay δ for different UAV velocity. Both the source and the UAV are assumed to transmit with a constant power P , with P setting to a value so that the average received signal-to-noise ratio (SNR) at the UAV for the static relaying is 10 dB. Note that the direct link between source and destination is assumed to be blocked and thus ignored. For simplicity, we assume that the Doppler effect due to the UAV's mobility has been well compensated. It is observed that for sufficiently high delay tolerance δ , the mobile relaying strategy achieves a throughput more than twice of that by static relaying. Furthermore, for any fixed δ , larger throughput is achieved for higher UAV velocity, which is as expected.

Note that an alternative strategy of mobile relaying is known as data ferrying or load-carry-and-delivery [5]. With this strategy, the UAV "loads" the data from the source as it reaches the nearest possible location from the source, flies towards the destination with the loaded data until it reaches the nearest possible location to the destination, and then delivers the data to the destination. As data ferrying has less communication time than the proposed mobile relaying, its achievable throughput is expected to be smaller, especially for cases with low UAV speed and/or stringent delay requirements. Furthermore, in the above discussions, a data buffer with sufficiently large buffer size is assumed at the UAV. In general, there exists a trade-off between on-board buffer size and achievable throughput in the mobile relaying design.

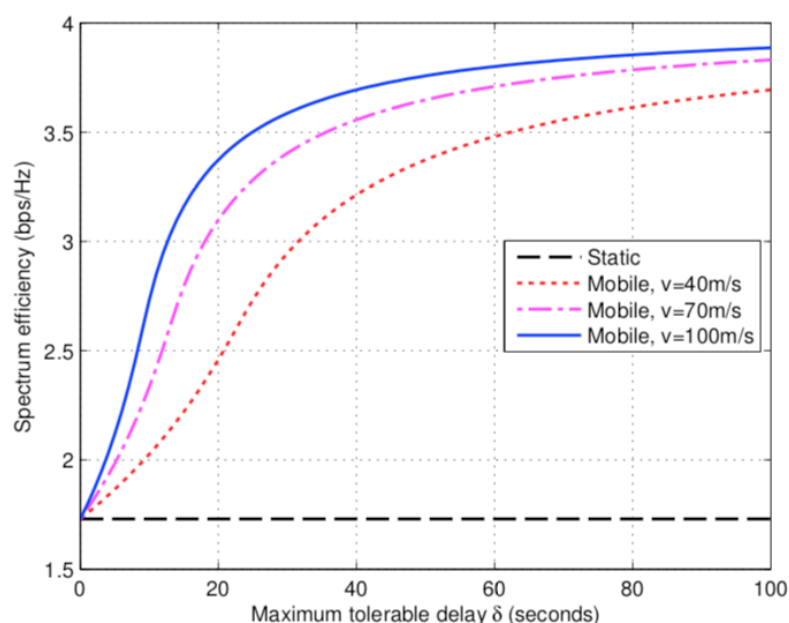


Fig. 4: Spectrum efficiency versus maximum tolerable delay with mobile versus static relaying.

B. D2D-Enhanced UAV Information Dissemination

D2D communication is an effective technique for capacity improvement in terrestrial communication systems [15]. The main idea is to offload the BS by enabling direct communications between nearby mobile terminals. For UAV-aided communication systems, D2D communication is expected to play an important role by providing the additional benefits such as UAV energy saving, lower capacity requirement for UAV wireless backhaul, etc. Many existing D2D techniques for terrestrial communication systems, such as those on interference mitigation and spectrum sharing, can be directly applied in UAV-aided communications, especially in the scenario to support ubiquitous cellular coverage as shown in Fig. 1(a). On the other hand, new D2D communication techniques could be devised by exploiting the unique characteristics of UAV-aided communications. In the following, we present one such technique, termed D2D-enhanced UAV information dissemination, which aims to achieve efficient information dissemination to a large number of ground nodes by exploiting both D2D communications and the UAV mobility.

As illustrated in Fig. 1(c), we consider the scenario where one UAV flies over a certain area to distribute a common file to a large number of ground nodes. One simple approach to achieve this is by letting the UAV repeatedly transmit the same file as it flies over different ground nodes, until all of them successfully receive the file. It is not difficult to see that such a scheme requires substantial UAV retransmissions, and its performance is essentially limited by the ground terminals that experience the weakest channel conditions with the UAV. The D2D-enhanced information dissemination scheme can effectively solve this problem with a two-phase protocol, as illustrated in Fig. 5.

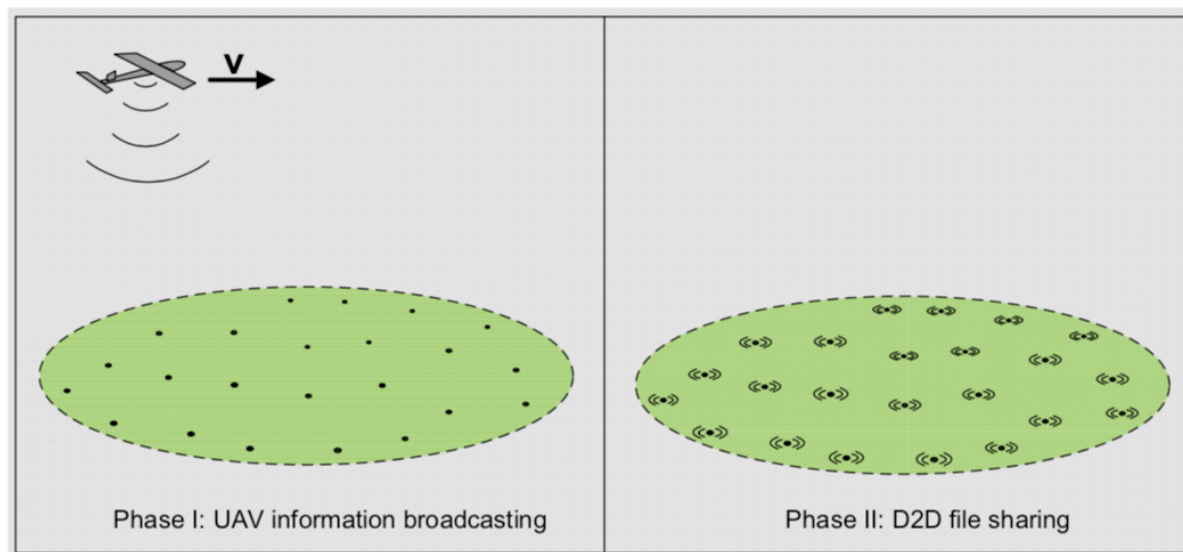


Fig. 5: The two-phase protocol of D2D-enhanced UAV information dissemination.

In the first phase, the UAV broadcasts the appropriately coded file to the ground nodes as it flies over them. Since each node has only limited wireless connectivity with the UAV, it is very likely that it can only successfully receive a fraction of the file, where different portions of the file are received by different nodes. In the second phase, the ground nodes exchange their respectively received data via D2D communications, until all the nodes receive a sufficient number of packets to successfully decode the file. This scheme significantly reduces the number of UAV retransmissions and as a result the total flying time of the UAV, which saves its energy and is particularly useful for small UAVs with limited

onboard energy. Notice that if the ground nodes are distributed over a wide geographical area, efficient node clustering algorithms can be applied, to improve the file sharing performance of short-range D2D communications within each cluster. The joint optimization of the UAV path planning, coding, node clustering, as well as D2D file sharing for this scenario is an important problem for future research.

6. CONCLUSIONS

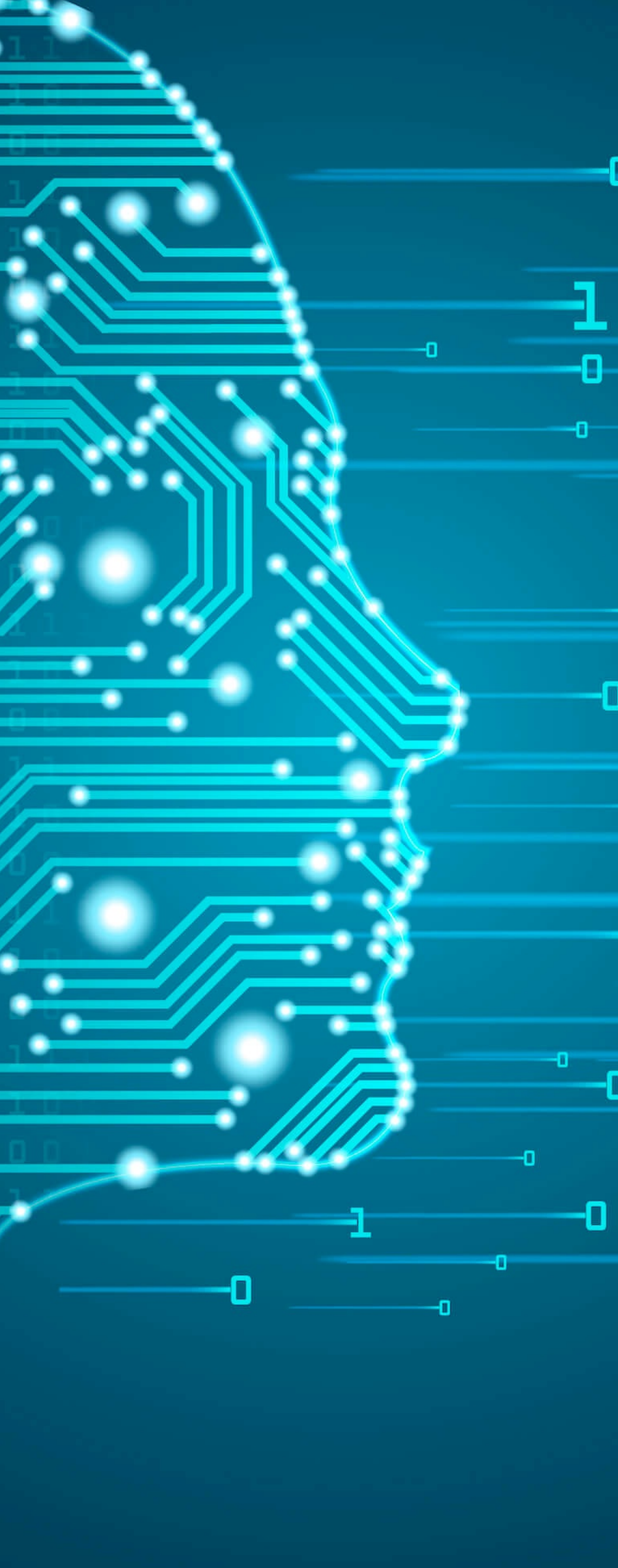
In this article, we have provided an overview on UAV-aided wireless communications with the help of three use cases: UAV-aided ubiquitous coverage, UAV-aided relaying, and UAV-aided information dissemination. The basic networking architecture and main channel characteristics were introduced. Furthermore, the key design considerations for UAV communications were also discussed. Lastly, we highlighted two key performance enhancing techniques by utilizing the UAV controlled mobility, including UAV-enabled mobile relaying and D2D-enhanced UAV information dissemination. It is hoped that the challenges and opportunities described in this article will help pave the way for researchers to design and build UAV-enhanced wireless communications systems in the future.

REFERENCES:

1. K. P. Valavanis and G. J. Vachtsevanos, *Handbook of unmanned aerial vehicles*. Springer Netherlands, 2015.
2. US Department of Transportation, “Unmanned aircraft system (UAS) service demand 2015–2035: Literature review & projections of future usage,” Tech. Rep., v.0.1, DOT-VNTSC-DoD-13-01, Sep. 2013.
3. A. Merwaday and I. Guvenc, “UAV assisted heterogeneous networks for public safety communications,” in *Proc. IEEE Wireless Commun. Netw. Conf.*, pp. 329–334, 9–12 Mar. 2015.
4. A. Osseiran et al., “Scenarios for 5G mobile and wireless communications: the vision of the METIS project,” *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 26–35, May 2014.
5. E. W. Frew and T. X. Brown, “Airborne communication networks for small unmanned aircraft systems,” *Proc. IEEE*, vol. 96, no. 12, pp. 2008–2027, Dec. 2008.
6. N. Goddemeier, K. Daniel, and C. Wietfeld, “Role-based connectivity management with realistic air-to-ground channels for cooperative UAVs,” *IEEE J. Sel. Areas Commun.*, vol. 30, no. 5, pp. 951–963, Jun. 2012.
7. D. W. Matolak and R. Sun, “Unmanned aircraft systems: air-ground channel characterization for future applications,” *IEEE Veh. Technol. Mag.*, vol. 10, no. 2, pp. 79–85, Jun. 2015.
8. T. S. Rappaport, R. W. Heath Jr, R. C. Daniels, and J. N. Murdock, *Millimeter wave wireless communications*. Prentice Hall, 2014.
9. R. Sun and D. W. Matolak, “Initial results for airframe shadowing in L- and C-band air-ground channels,” in *Proc. Integrated Commun., Navigation, and Surveillance Conf.*, pp. 1–8, Apr. 2015.
10. Z. Han, A. L. Swindlehurst, and K. J. R. Liu, “Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3533–3546, Sep. 2009.
11. T. Schouwenaars, B. D. Moor, E. Feron, and J. How, “Mixed integer programming for multi-vehicle path planning,” in *Proc. European Control Conf.*, pp. 2603–2608, 2001.

REFERENCES:

12. A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.
13. M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *Proc. IEEE Global Telecom. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2015.
14. F. Bohagen, P. Orten, and G. E. Oien, "Design of optimal high-rank line-of-sight MIMO channels," *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, pp. 1420–1425, Apr. 2007.
15. A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1801–1819, Apr. 2014.



Integration of Machine Learning to simplify the Analysis in Security Operations Center (SOC)

Chirath De Alwis



ABOUT THE AUTHOR

CHIRATH DE ALWIS

Chirath De Alwis is an experienced information security professional with more than five years' experience in Information Security domain. He holds MSc IT (Specialized in Cyber Security), PGdip IT (Specialized in Cyber Security), BEng (Hons) Computer network and Security (UK) and six professional certifications. Currently, Chirath is involved in vulnerability management, incident handling and digital forensics activities in Sri Lankan cyberspace

Introduction

Machine learning has become a buzz word in the recent past. Therefore, it is common that most often this term is misused. Machine learning (ML) is an algorithm that gives the software applications it is applied to the ability to autonomously learn from its own environment, then improve operations based on the data collected [1]. Machine learning can be categorized into two parts;

- Supervised:

In this approach the model is fed with a data set that is well labeled [2]. This data labeling helps the model to understand what is correct and what is wrong.

- Unsupervised:

In this approach the model is fed with a data set that is not labeled. Instead of feeding the model with already labeled data, this method allows the model to work on its own to discover information [2].

When it comes to security operations, this machine learning can play a huge role in simplifying the analyst's tasks. But in order to get the advantage of using machine learning technology it is required to have an understanding of both data science and security operations.

Integrating Machine Learning into SOC operations

Integration of machine learning into SOC operations can be categorized into three main stages. Those are [3];

- Prevention and Detection
- Incident Response
- SOC Management

Machine Learning for Prevention and Detection

Attack prevention and detection is one of the major responsibilities that a security analyst has. Incorporating the existing vulnerability management, attack detection and monitoring tools with cyber threat intelligence helps to build a machine learning platform that can provide not only attack prediction but also to perform automated threat hunting against the infrastructure. The ability to continually and dynamically learn what's normal in behavior allows analysts to understand advanced threats.

A recent study [4] was conducted to create a user-centric machine learning framework to identify risky users from the network. Due to the large number of alerts generated from Security Information and Event Management (SIEM) tools,

it is difficult to understand false positives. Being the majority of SIEM generated alerts are false positives, there is a time waste when the analyst analyzes these false positives. This proposed solution creates the features based on individual users; some of the common features used in this solution are number of alerts per day and event arrival rate. The data labeling was performed in this solution using the analyst results. This solution demonstrates that it is able to learn more insights from the data with highly unbalanced and limited labels [4]. This allows automatic identification of false positives when the alert is triggered and if the alert is false positive, the alert will be forwarded to the analyst to conduct further analysis.

A similar study was conducted [5] to analyze the alerts generated in monitoring devices and identify false positives from the alert pipeline. This approach detects alerts from various sources and based on the analyst's feedback on similar alerts it detects the false positives. Once a signature-based detection system detects the alert from various sources and sends it to the analysis pipeline, these alerts will be passed through a machine learning model that classifies the alerts as "threat" or "false positive". Based on the classification, the alerts will be parented to the analysts to perform further analysis [5]. The following image depicts the workflow of this proposed system.

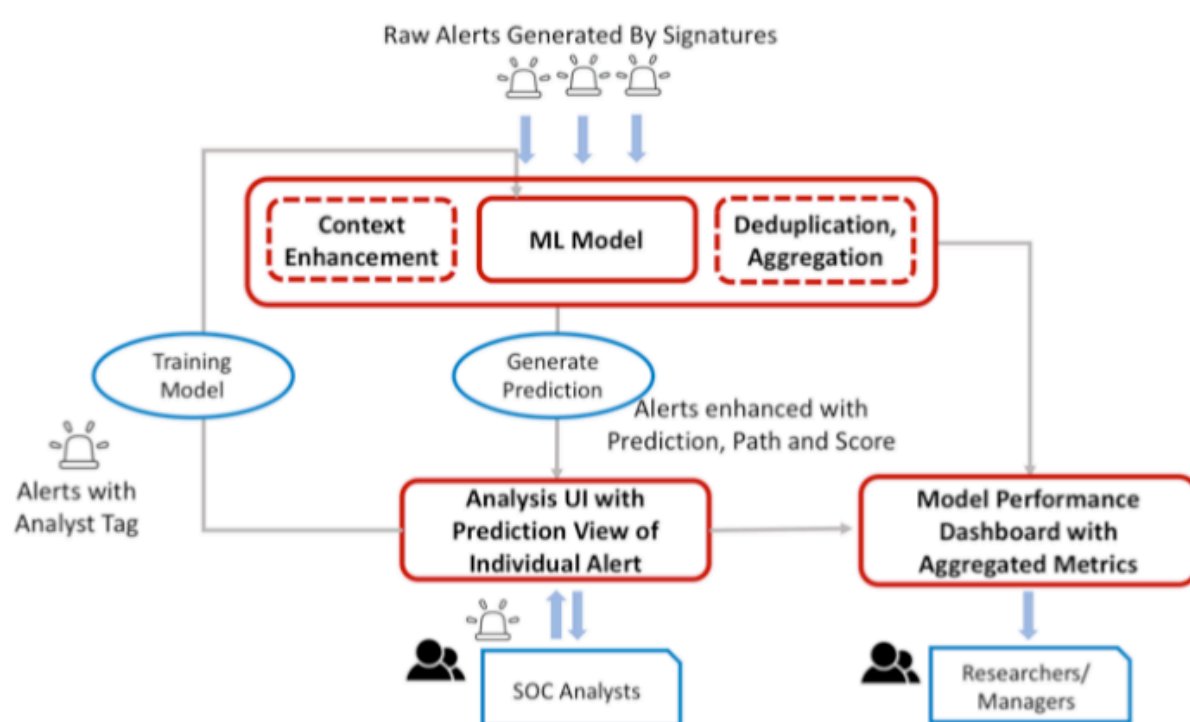


Figure 1: System Workflow [5]

Integration of MITRE ATT&CK framework [6] with using machine learning into security operation center is another use of this technology. Once this framework is integrated and automated with machine learning, it is possible to automatically detect chained attacks as well as alert correlation within the network. This allows SOC analysts to perform automated threat hunting activities to detect advanced persistent threats from the enterprise network. These activities increase the detection capabilities within the security operations center while reducing the time and effort put by physical analysts. This increases the efficiency within the SOC operations and allows analysts to perform various other tasks. Most of the modern security tools such as CrowdStrike [7] and Checkpoint [8] provide this capability with their product line.

Machine Learning for Incident Response

When an incident occurs, it involves some effort from SOC analysts to identify and analyze multiple data points to identify the scenario and provide first responder procedure. Analysis of these data points can be automated using machine learning that will look for potential data points and provide the initial analysis to the analyst. As an example, machine learning can be applied to illustrate similar anomalies that have arisen previously, thereby shortening the analyst's investigation time, giving them important points of reference, and even potentially deploying the proper incident response playbook [3]. This allows analysts to conduct first responder procedure with a very limited amount of time.

Research was conducted [9] to perform automated incident categorization using machine learning. Researchers have analyzed the past incident and, based on categorization of past incidents by analyzing incident description, they have come up with a machine learning module that will analyze the incident description and, based on the description and past incidents, the trained model is capable of categorizing an incident. A similar approach can be performed to categorize the potential impact and urgency to benchmark security incidents.

Machine Learning for SOC Management

Machine learning can be integrated not only to analyze the activities that flow through the security operations center but also to analyze activities of security operations centers. One critical area where machine learning can be useful in SOC management is to analyze the number of security incidents, intelligence and alerts based on the dates and time and forecast security incidents. This allows management to prepare for upcoming threats and work proactively. Based on this information the management can adjust their shifts as well. As an example, it is common that there is a high possibility of getting attacks and alerts during holidays because most of the adversaries will target these holidays to perform their attacks. Therefore, as management, it is possible to adjust the shifts to make sure when such an incident happens, at least one or two members are available to address the issue.

Machine learning will also be helpful in making some management decisions such as planning for next year's security budget and identifying areas where they need to improve or strengthen. As an example, an organization may receive a high number of alerts related to malicious file download from end users. This information indicates that the organization needs to strengthen their end users and increase awareness on security. Another organization may receive a considerable amount of DOS and DDOS attacks. This is an indication to the management to invest on DOS and DDOS prevention solutions to prevent such attacks.

Conclusion

Machine learning is a technology that helps security operations centers to address some critical areas and save analyst's time and increase efficiency. In order to get the maximum utilization of machine learning, it requires some

prior expertise in data science. If an organization does have an inbuilt data science team, this technology allows organization to reduce the cost for investing in expensive tools out there in the industry.

Reference:

1. Help Net Security. 2020. Machine Learning Trumps AI For Security Analysts - Help Net Security. [online] Available at: <<https://www.helpnetsecurity.com/2019/01/21/machine-learning-trumps-ai-for-security-analysts/>> [Accessed 22 March 2020].
2. Guru99.com. 2020. Supervised Vs Unsupervised Learning: Key Differences. [online] Available at: <<https://www.guru99.com/supervised-vs-unsupervised-learning.html>> [Accessed 22 March 2020].
3. Siemplify. 2020. What Machine Learning Means For Security Operations | Siemplify. [online] Available at: <<https://www.siemplify.co/blog/what-machine-learning-means-for-security-operations/>> [Accessed 22 March 2020].
4. C. Feng, S. Wu and N. Liu, "A user-centric machine learning framework for cyber security operations center," *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Beijing, 2017, pp. 173-175. URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8004902&isnumber=8004858>. [Accessed 22 March 2020].
5. Awalin, S., Matthew, B. and Murali, M., 2020. Building A Machine Learning Model For The SOC, By The Input From The SOC, And Analyzing It For The SOC - IEEE Conference Publication. [online] ieeexplore.ieee.org. Available at: <<https://ieeexplore.ieee.org/document/8709231>> [Accessed 22 March 2020].
6. Attack.mitre.org. 2020. MITRE ATT&CK®. [online] Available at: <<https://attack.mitre.org/>> [Accessed 22 March 2020].
7. crowdstrike.com. 2020. CrowdStrike: Cloud-Native Endpoint Protection Platform. [online] Available at: <<https://www.crowdstrike.com/>> [Accessed 22 March 2020].
8. Check Point Software. 2020. Preventing Zero Day Attacks Using MITRE ATT&CK Framework - Check Point Software. [online] Available at: <<https://blog.checkpoint.com/2020/01/20/preventing-zero-day-attacks-using-mitre-attck-framework/>> [Accessed 22 March 2020].
9. Silva, S., Ribeiro, R. and Pereira, R., 2020. Machine Learning In Incident Categorization Automation - IEEE Conference Publication. [online] ieeexplore.ieee.org. Available at: <<https://ieeexplore.ieee.org/document/8399244>> [Accessed 22 March 2020].