# Wi-Fly? : Detecting Privacy Invasion Attacks by Consumer Drones

Simon Birnbach
University of Oxford
simon.birnbach@cs.ox.ac.uk

Richard Baker
University of Oxford
richard.baker@cs.ox.ac.uk

Ivan Martinovic
University of Oxford
ivan.martinovic@cs.ox.ac.uk

***Abstract*—Drones are becoming increasingly popular for hobbyists and recreational use. But with this surge in popularity comes increased risk to privacy as the technology makes it easy to spy on people in otherwise-private environments, such as an individual's home. An attacker can fly a drone over fences and walls in order to observe the inside of a house, without having physical access. Existing drone detection systems require specialist hardware and expensive deployment efforts; making them inaccessible to the general public.**

**In this work we present a drone detection system that requires minimal prior configuration and uses inexpensive commercial off-the-shelf (COTS) hardware to detect drones that are carrying out privacy invasion attacks. We use a model of the attack structure to derive statistical metrics for movement and proximity, that are then applied to received communications between a drone and its controller. We tested our system in real world experiments with two popular consumer drone models mounting privacy invasion attacks using a range of flight patterns. We were able to both detect the presence of a drone and identify which phase of the privacy attack was in progress. Even in our worst-case we detected an attack before the drone was within 48m of its target.**

## I. Introduction

### A. Motivation

Small unmanned aerial vehicles (UAVs) are becoming increasingly commonplace nowadays. Since the release of the Parrot AR.Drone in 2010, the use of these devices is no longer restricted to military and commercial domains nor enthusiasts, but has opened up to laymen as well. Advances in sensor design and image processing support sophisticated flight-assist features that make piloting UAVs easily accessible to hobbyists. This is further supported by the ubiquity of smartphones and tablets, as these devices make convenient controllers for consumer models. As consumers get a broader choice with more affordable products coming out every year, sales of UAVs for videography and recreational use are rising. Most such UAVs are multirotor aircraft that use four or more rotors to fly and are able to hover in mid-air.


NDSS '17, 26 February - 1 March 2017, San Diego, CA, USA
 ISBN 1-1891562-46-0
http://dx.doi.org/10.14722/ndss.2017.23335

The retail tracking service NPD Group has reported that consumer UAV sales have grown by 224% to almost $200 million in the time from April 2015 to April 2016 and that the sales growth was accelerating over this period [14]. According to the US American Federal Aviation Administration (FAA), 1.9 million small UAVs, more commonly known as drones, were sold in 2016 on the US hobbyist market alone. The FAA predicts that the number of annual sales will increase to 4.3 million units by 2020 [2].

But this prevalence of consumer UAVs is not without risks. Over the past few years incidents at high-security facilities have occurred with increasing frequency. Prisons have to deal with drones dropping weapons and other contraband into prison yards [8], drug smugglers use them to carry their goods over borders [9], and the American Secret Service repeatedly had to deal with people who managed to bypass security measures and crashed their drone on the White House lawn [22], [21]. Besides the danger that drones pose to restricted areas, there has been an increasing unease in the general population about privacy invasion through drones carrying high-fidelity camera equipment. As most consumer models are outfitted with cameras for live-view video during flight, UAVs can fly over fences to see into nearby gardens or even look into windows to observe the interior. In Seattle, a woman called the police after spotting a drone flying in front of her 26th-floor apartment window; observing her partially-undressed inside [23]. Meanwhile, a man from Kentucky was arrested after shooting down a drone over his property. He explained this act by saying that the drone was spying on his sunbathing daughter [29].

These and similar incidents have prompted reactions from regulators, such as the FAA requiring since December 2015 that drone owners register as drone operators in order to fly UAVs legally [3]. It has also led to a multitude of defence mechanisms and detection systems being proposed (see Section I-B for an overview of existing methods), however these systems are mostly targeted at law enforcement and larger companies. By contrast, we focus on threats to private individuals. An attacker can launch a drone from a distant location and fly it to an otherwise-inaccessible private area, such as outside a top-floor window within a walled garden, and freely observe the occupants. The occupants require a system that alerts them to the presence of an impending privacy invasion, with sufficient warning to allow them to react accordingly. The system must be easy to deploy without specialist knowledge and detect a wide variety of drones based on intrinsic properties of UAV activity; both to be robust to the

rapid pace of advancement in drone technology and to avoid spurious alarms.

In this work, we propose such a system that detects the presence of a consumer UAV with live-video streaming being used to invade someone's privacy in their home. We make use of readily-available commercial, off-the-shelf (COTS) hardware to measure the signal strength of the communication between the drone and its controller; developing metrics that identify signal properties inherent to drone flight. In doing so we make the following specific contributions:

- Development of a model of UAV-based privacy-invasion attacks
- Derivation of statistical metrics to identify drone movement and proximity
- Proposal of mechanisms to overcome attempts by an attacker to avoid detection, such as varying their speed or flight pattern
- Reporting the evaluation of our system in a real-world scenario using popular consumer drones

### B. Related work

In this section, we will first survey existing UAV detection methods and subsequently give a short overview over available defence mechanisms.

A traditional way to detect aircraft is through the use of radar; a staple of military and aircraft control for a long time. But conventional radar systems are not able to detect drones. To detect small objects like drones, expensive high-frequency radar systems are necessary [11]. However, Boddhu et al. [5] suggest that drones might be indistinguishable from birds for such a radar system as they have a comparable wing span. They propose an approach that uses humans as sensors by building a collaborative smartphone app that allows users to share drone sightings. Their approach is more appropriate to target large-scale threats, and is not applicable for the defence of a single property against a nearby attacker.

Another way to detect drones is based on video cameras and image analysis. Rozantsev et al. [19] detect small UAVs by using both their appearance and motion cues. According to the work by Busset et al. [6] the variety of drone shapes is challenging for appearance-based approaches, whereas methods based on motion cues struggle with similarities between drone and bird movement. Therefore, they propose using acoustic cameras to complement the use of video cameras. These microphone arrays use the noise of the rotors to detect drones. As they use specialised and expensive equipment for their system, this is not a feasible solution for the use in a domestic setting. Case et al. [7] designed a low-cost acoustic array to detect small UAVs that can be built with COTS hardware. But their system is also not easy to build and deploy as it requires an array of 24 microphones. Furthermore, the total cost of their system still amounts to $3768 which makes it unaffordable for the general public.

Hybrid approaches such as CSUAV [28] combine radar, acoustic arrays and video cameras to profit from the benefits of all approaches. But while they improve the detection performance, they also increase the complexity and cost of the system substantially.

| Detection method | Drone Detector[3] | Drone Shield[4] | Dedrone[2] | Orelia Drone Detector[5] | Domestic Drone Countermeasures[6] |
|---|---|---|---|---|---|
| Audio | Yes | Yes | Yes | Yes | No |
| Video | No | No | Yes | No | No |
| RF | Yes | No | Yes | No | Yes |

TABLE I: Commercial drone detection systems. Comparison based on [10] and respective product feature descriptions.

Most of the aforementioned approaches are focused on the protection of high-security facilities; permitting the use of expensive, specialist equipment, which is infeasible in a domestic setting. As we are targeting such an environment, we need to be able to use cheap COTS hardware. That is why we focus on using the radio frequency (RF) signal of the drone which can be detected using inexpensive consumer products. The localisation of RF signals in general is a well studied topic. Localising drones in 3D space would allow one to detect them as flying transmitters. However, these methods usually need several receivers and often rely on expensive equipment and precise synchronisation between the receivers (see [20] for an overview of signal localisation methods).

The academic work on UAV detection with more affordable methods is quite limited. Peacock and Johnstone [18] detect drones by using protocol signatures from the drone's Wi-Fi connection. For this method to work, they have to rely on an unencrypted connection between drone and controller, which is not the case for newer models. They also discuss the use of media access control (MAC) address prefixes by known manufacturers to recognise transmitting drones. Although this is an easy and reliable method to detect manufacturers that have their own MAC prefix range, it is only capable of detecting known drone models. As there is an ever-increasing variety of drone models, it gets more difficult to build and update a comprehensive database of MAC prefixes. Furthermore, some manufacturers use MAC prefixes that are not specific to them, e.g., if their camera system consists of the popular GoPro cameras[1] which are widely used on their own. But even when the MAC prefix of a drone manufacturer is detected, relying only on the presence of certain MAC addresses is not enough to distinguish a neighbour turning on a drone in their house from an actual privacy-invasion attack.

Over the last few years, several companies have entered the market for drone detection systems. With the exception of Domestic Drone Countermeasures, their intended target groups are law enforcement agencies and security forces of larger companies. See Table I for a comparison of their detection methods. Domestic Drone Countermeasures use a mesh network of receivers to establish the mobility of transmitters and treat all unidentified moving transmitters as a threat. Our goal in this work, on the other hand, is it to detect an actual privacy invasion attack using only one inexpensive receiver.

In our scenario, establishing line-of-sight to the receiver plays an important role. Xiao et al. [30] studied the detection of

[1] www.gopro.com

[2] www.dedrone.com
[3] www.dronedetector.com
[4] www.droneshield.com
[5] www.drone-detector.com
[6] www.ddcountermeasures.com

line-of-sight/non-line-of-sight (LOS/NLOS) conditions. However, they are focused on an office setting with tighter control over the transmitters and most of their metrics rely on such an indoor environment. Furthermore, the transmitters are static in this work, whereas mobility is an intrinsic feature of drones. The detection of moving transmitters has been investigated by Muthukrishnan et al. [17] which we will revisit in Section V.

Once a drone has been detected acting maliciously, some set of countermeasures may be deployed. A vast array of countermeasures have been proposed; from shotguns, through flying nets [26], live eagles [1], control-signal jamming [13], Wi-Fi de-authentication attacks [18], flight-controller exploitation [15] and GPS spoofing [24], [13], to interference with onboard gyroscopes via acoustic resonance [25]. In a domestic setting, the course of action could be as simple as automatically shutting the blinds. The relative merits of each approach warrant careful consideration, however they are beyond the scope of this work and we do not discuss countermeasures further — focusing solely on the mechanics of detection.

## II. Background

The market for consumer drones is contested by many different manufacturers, but at time of writing it is dominated by only a few companies, namely DJI Innovations (49% US market share), Parrot (19%), Protocol (6.3%), Yuneec (5.6%) and 3D Robotics (4%) [27][7].

The majority of drones provide a live video stream back to the operator; so-called "first-person view" (FPV). In all but the simplest of flights it is a great benefit to the pilot to be able to see from the perspective of the drone itself. The user can commonly connect to the drone with their smartphone, tablet or a dedicated controller to receive the video, and often record the footage in higher resolution as well. Interoperability with existing user devices has played a great role in enhancing the appeal of these consumer UAVs and hence most drones use the common 2.4GHz or 5.8 GHz Wi-Fi bands for their video downlink. Some even use Wi-Fi for the telemetry (control channel) of the drone. In these cases, the drone can be controlled completely by an app. Otherwise, a separate, dedicated controller is needed and the Wi-Fi connection is only used for the live-view video. Table II shows different models from different manufacturers and the communication technology used for their video downlink. With the exception of newer DJI drones, which use a proprietary technology named "Lightbridge", Wi-Fi is at least optional in all cases, if not the only way to receive video during flight. As such we consider Wi-Fi as the drone communication system throughout this work.

Airspace regulations differ between jurisdictions in terms of the requirements placed upon drone operators. Almost all mandate that minimum separation distances are observed between the drone and any surrounding persons or property. In the United Kingdom, the Civil Aviation Authority (CAA) classes drones equipped with cameras as "small unmanned surveillance aircraft" and requires that they [4]:

- fly no closer than 50m from any persons, buildings or vehicles that are not under the control of the operator
- stay below a 400ft flight ceiling
- stay within line-of-sight of the operator

| Brand | Model | Video Downlink | Speed (m/s) |
|---|---|---|---|
| DJI[8] | Phantom 3 Standard | Wi-Fi (2.4 GHz) | 16 |
| | Phantom 3 Advanced/Pro | Lightbridge | 16 |
| | Phantom 4 | Lightbridge | 20 |
| Parrot[9] | AR.Drone 2.0 | Wi-Fi (2.4 GHz) | 11.11 |
| | Bebop | Wi-Fi (2.4, 5.8 GHz) | 13 |
| | Bebop 2 | Wi-Fi (2.4, 5.8 GHz) | 18 |
| Protocol[10] | Dronium One WiFi Ed. | Wi-Fi (2.4 GHz) | N/A |
| Yuneec[11] | Typhoon H | Wi-Fi (5.8 GHz) | 13.5 |
| | Tornado H920 | Wi-Fi (5.8 GHz) | 11.11 |
| 3D Robotics[12] | Solo | Wi-Fi (2.4 GHz) | 24.6 |
| | IRIS+ | Wi-Fi optional | 22.7 |
| | X8+ | Wi-Fi optional | 30 |

TABLE II: Features of popular drones with live-view video.

## III. Adversary model

We consider an adversary that operates an unmodified, commercially-available drone with the intent of invading the privacy of their neighbours. The attacker tries to capture video of the interior of a building that they would not be able to see into otherwise, against the will of the inhabitants. The attacker is considered to be purely an opportunist acting inappropriately with standard equipment; they are not capable of modifying the drone in any way.

The adversary does not have access to the premises and is thus positioned some distance away from the target window. To carry out the attack, the adversary has to move the drone towards the window until it is in line-of-sight (LOS) of the window and close enough that they can observe the interior of the building by using the drone's onboard camera. However, they cannot get arbitrarily close due to turbulent air patterns from the facia of the building making the drone harder to control and the increased risk of detection and physical interception by inhabitants of the building. The quality of the footage is determined not only by the distance, but also the camera resolution and the field-of-view (FOV) that the window allows. The speed of the approach is limited only by the drone's capabilities.

We identify three phases of the attack:

*a) Approach:* The drone is launched and approaches the window until it is close enough for surveillance. In doing so it must establish line-of-sight (LOS).

*b) Surveillance:* While hovering in front of the window, the drone records video footage of the interior. The movement of the drone in this phase is kept minimal in order to increase the quality of the recorded footage.

*c) Escape:* After successfully surveilling the interior of the house, the drone moves away from the window and returns to the launch site.

While the attacker cannot alter the fundamental operation of their drone, they can vary the flight pattern and speed of the

[7] Based on data provided by the NPD Group/Retail Tracking Service
[8] www.dji.com
[9] www.parrot.com
[10] www.protocolrchelicopter.com
[11] www.yuneec.com
[12] 3dr.com

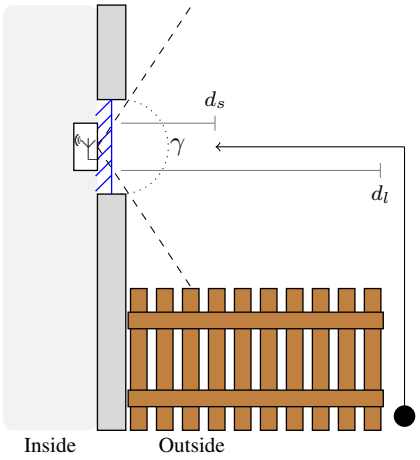



Fig. 1: The attacker launches the drone at launch distance $d_l$ and flies it to surveillance distance $d_s$ to carry out the attack. The detector is installed in the window. The FOV $\gamma$ of the window limits the area that is in LOS of the detector.

drone in an attempt to avoid detection. For example, they may try to approach as quickly as possible, as slowly as possible; attempt to mimic hobbyist flight or stay out of sight from the window for as long as they can. The attacker is limited only by the bounds of the drone's manoeuvrability.

## IV. System model

We model the system as follows; visualised in Figure 1. As the adversary does not have access to the premises of the defender, there is a minimum launch distance $d_l$ between them and the target window. The adversary moves the drone towards the window with any speed up to a maximum of $v_{\max}$, representing an upper bound on the speed of contemporary drones. As the drone progresses towards the window, the separation reduces to a surveillance distance $d_s$, at which the privacy invasion can take place. As mentioned in Section III, the attacker must balance the risks to their equipment and successful attack against the quality of the captured footage, and adopt a value of $d_s$ such that they do so. The attacker must also establish LOS with the target window; the limits upon this are the FOV of the window, taken as $\gamma$.

A small Wi-Fi receiver is mounted in the window. The receiver is configured to capture traffic in monitor mode, periodically switching channels to cover the entire band.

The received traffic is separated into flows, filtered and analysed to determine first whether an individual flow represents a live-streaming drone and then, if so, whether the drone is conducting a privacy invasion attack. The system maintains averages for the received signal strength (RSS) variance (in the form of standard deviation) of each flow, over rolling time windows of various sizes. It also monitors the baseline RSS variance of packets from a known, static transmitter (such as the user's accesspoint).

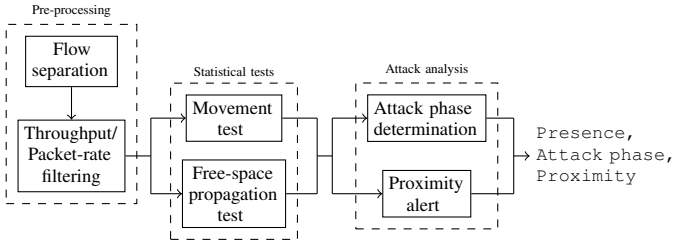



Fig. 2: Flow diagram of the detection algorithm.

## V. Detection

Detection proceeds in three broad phases. The *pre-processing* phase prepares the stream of received packets for analysis. Then *statistical tests* are applied to establish whether individual transmitters are moving and operating in free space. If the statistical tests pass then the *attack analysis* phase determines the attack phase that is in progress and whether the drone has reached close proximity. The steps are described below and shown in Figure 2.

### A. Pre-processing

The first step towards detection is the separation of different data flows. Given the assumption that the drone is unmodified and that communication makes use of the IEEE 802.11 Wi-Fi standard, flows are separated by MAC address. The provision of a continuous, live video stream from the drone to the controller, necessitates a consistent, high bandwidth utilisation on the communication channel[13]. The detection process excludes flows that do not display this characteristic.

### B. Statistical tests

An airborne drone operates largely in free space, and must establish LOS to the window (and thus the detector) in order to conduct a privacy invasion attack. As such the transmission environment is uncluttered and the received signal strength (RSS) can be expected to be dominated by the direct-path component. We neglect multipath effects due to nearby houses and the ground as the drone has this strong, short-range LOS connection. See Figure 3 to see how the path loss changes over time for an attack with a straight approach and constant speed under the free-space propagation assumption.

The system applies statistical tests to a flow to establish whether it is likely to represent a drone or not. The tests are based upon a comparison of the standard deviation of RSS over short and long rolling time windows.

Given the high packet rate exhibited by a live-streaming drone, in a sufficiently short period the drone does not move enough for the change in RSS due to movement to be distinct from noise. For a non-moving LOS transmitter in a static environment, changes in signal strength are largely due to measurement noise and cross-traffic interference. Hence, we expect a standard deviation that is close to that of the general noise level in such a timeframe. This noise threshold $\sigma$ represents the standard deviation of the noise the receiver is subject

[13]For example, a Parrot Bebop drone creates 400 packets/s, for a throughput of 500KB/s, when providing 720p video at 30 frames-per-second.

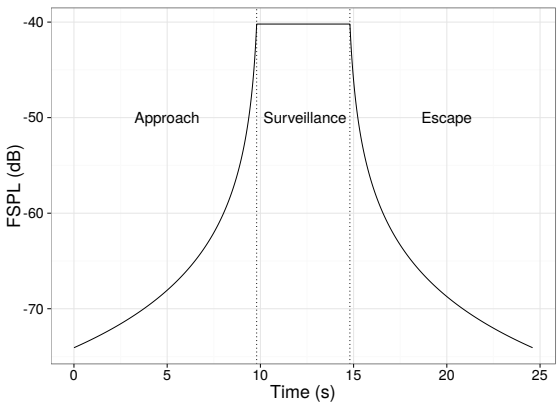



Fig. 3: Simulation of the Free Space Path Loss during a privacy invasion attack with $5s$ surveillance. The speed of the drone is $5m/s$, the surveillance distance $d_s$ is $1m$.

to. It can be determined in an automatic calibration phase by measuring the baseline noise from a known transmitter in a static environment. As an example, the accesspoint of the user can be used to calibrate the system.

Over a long time period, however, the RSS from a moving drone is mainly affected by that movement. So, whilst the standard deviation alone may not be enough to confirm LOS transmission[30], we can additionally use the long-term changes that are detectable by having a higher standard deviation than for immobile transmitters [17].

In combination, a low short-term standard deviation together with a high long-term standard deviation suggests a LOS transmitter that is moving in free-space. We expect other moving transmitters on the ground or indoors to have more short-term variation due to changing multipath effects. Whereas immobile transmitters in a static environment are expected to have a stable long-term standard deviation, while those in a dynamic environment are expected to experience substantial short-term changes.

In order to apply both tests, the system must first determine appropriate short and long window sizes. To find a window size where the change in signal strength stays below the noise level, we consider the part of the approach with the maximal possible change. This change occurs when the drone travels the final stretch at maximal speed before it arrives at surveillance distance $d_s$[14].

We assume that the drone transmits with packet rate $r$, and $d_s$, $v_{\max}$ and transmission frequency $f$ are given. At packet rate $r$ we receive $N = w \cdot r$ transmissions in a time window of length $w$. Then the free-space path loss of each of these $N$ measurements is given by [12]:

$$x_i = 20\log_{10}\left(d_s + \frac{i}{r}\cdot v_{\max}\right) + 20\log_{10}(f) - 27.55$$

[14]This is unlikely to happen in reality, as it is too hard to control a UAV at that speed so close to the window. Nevertheless, it helps to determine the largest change that is physically possible and caused by movement alone.

The unbiased sample standard deviation can be computed with:

$$s(N) = \sqrt{\frac{1}{N-1}\sum_{j=1}^{N} x_j - \bar{x}}$$

where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ is the sample mean.

We then have to compute the maximal window size for which the standard deviation is below the noise threshold $\sigma$:

$$w_s = \max\{w | s(w \cdot r) < \sigma\}$$

Then $\sigma$ bounds the standard deviation of the random variable $FSPL$ within window $w_s$.

We consider the signal strength as the sum of random variables $FSPL$ for the free-space path loss and $X_N$ for the noise. Then the variance of the sum of these random variables is given by [16]:

$$Var(FSPL + X_N) = Var(FSPL) + Var(X_N) + 2Cov(FSPL, X_N)$$

We know from above that $Var(FSPL) < \sigma^2$ and $Var(X_N) = \sigma^2$. Additionally, we know that $FSPL$ and $X_N$ are uncorrelated, as they are independent random variables. This means that $Cov(FSPL, X_N) = 0$ and it follows that:

$$Var(FSPL + X_N) < \sigma^2 + \sigma^2 + 2 \cdot 0 = 2\sigma^2$$

Consequently, the short-term *free-space propagation* test fails when the standard deviation of the measured samples during $w_s$ surpasses $\sqrt{2}\sigma$.

Analogously, we compute the larger window size to detect movement by choosing an expected minimal velocity $v$ and by looking at the first stretch after launching at distance $d_l$. In this case, the free-space path loss for the $N$ corresponding measurements is:

$$x_i = 20\log_{10}\left(d_l - \frac{i-1}{r}\cdot v\right) + 20\log_{10}(f) - 27.55$$

This time, we are looking for the minimal window size for which the sample deviation is above the threshold:

$$w_l = \min\{w | s(w \cdot r) > \sigma\}$$

Again, using the independence of $FSPL$ and $X_N$ it follows that:

$$Var(FSPL + X_N) = Var(FSPL) + Var(X_N) > 2\sigma^2$$

As a result, the *movement* test is successful if the measured samples during $w_l$ have a standard deviation that is higher than $\sqrt{2}\sigma$.

### C. Attack analysis

All flows considered to be drones are examined to establish whether or not they are being used to mount a privacy invasion attack. The approach behaviour relative to the receiver is determined by monitoring the long-term RSS trend; whether it is increasing, stable or decreasing. A proximity test is then applied to drone flows that appear to be approaching or

hovering, to determine whether the drone has reached $d_s$ and begun surveillance.

The attack phase can be deduced by taking the difference between the mean of the first and the second half of $w_l$: $\bar{x}_{[1,\lfloor \frac{N}{2} \rfloor]} - \bar{x}_{[\lceil \frac{N}{2} \rceil, N]}$.

The sign of $\Delta x$ then determines the current attack phase; if the drone is approaching, $\Delta x$ is larger than zero, whereas values lower than zero indicate that it is escaping. If the sign of $\Delta x$ is zero, the drone is not moving. However, this is not sufficient to decide whether the surveillance phase has started. It could just be hovering at a larger distance from the window, without being able to observe the interior. Therefore, we have to take the *proximity* to the window into account.

We assume that the approach has been detected previously. Let $w_l$ be the window size in which the detection happened and let $v$ be the corresponding drone speed. Then the drone has arrived at surveillance distance $d_s$, if $\Delta x$ as defined above is larger than or equal to the following threshold $\sigma_p$:

$$x_i = 20\log_{10}\left(d_s + \frac{i}{r}\cdot v\right) + 20\log_{10}(f) - 27.55$$

$$N = w_l \cdot v,\ \ \Delta x \geq \sigma_p$$

One can see in Figure 8 that the window has to catch up to detect the surveillance phase. Therefore, it is important that an alarm is already raised if proximity to the window is detected after an approach phase. This is especially true, if the approach was very slow as $w_l$ might cover the whole surveillance period otherwise.

*Detection range*

To compute the time until our system can detect an attacker, we take $w_l$ as computed above: $t_d = w_l$.

Then the detection range $d_d$ is: $d_d = d_l - w_l \cdot v$. The detection range depends on $d_l$, whereas the movement speed $v$ has little effect on it, and only shows a slight variation due to the sampling rate.

As we do not know the actual speed of the attacking drone in advance, we have to compute a range of different window sizes in parallel to guarantee a timely detection.

*Approach patterns*

The case studied above corresponds to a direct approach to the window. As noted in Section III, the attacker can vary their approach in an attempt to avoid detection. We consider a set of example flight patterns; detailed in Table III and visualised in Figure 4. Whilst these patterns are not exhaustive, they demonstrate some extremes of the behaviour of an attacker. The first three patterns have in common that the attacker is in LOS of the window for most of the attack; the main difference between these three patterns is the effective speed of the drone towards the window. The fourth pattern, the NLOS approach, ensures that the drone is only in LOS of the window for a short period of time. By approaching from the sides, above or below like this, the attacker can give any detection approach, whether human or machine, less opportunity to detect the drone prior to surveillance beginning.

| Approach | Description | LOS |
|---|---|---|
| Direct | Drone follows shortest path to window | Constant |
| Zig-zag | Drone follows zig-zag pattern; resulting in variable effective approach speed | Constant |
| Back and forth | Drone approaches, backtracks, then progresses again; resulting in an effective approach speed that is sometimes negative | Constant |
| NLOS | Drone avoids LOS until shortly before surveillance phase | Emerges |

TABLE III: Example approach flight patterns

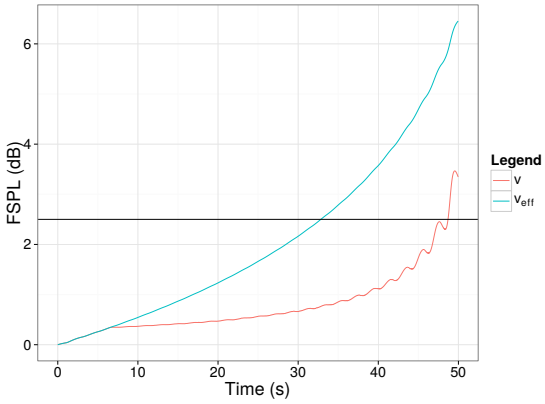
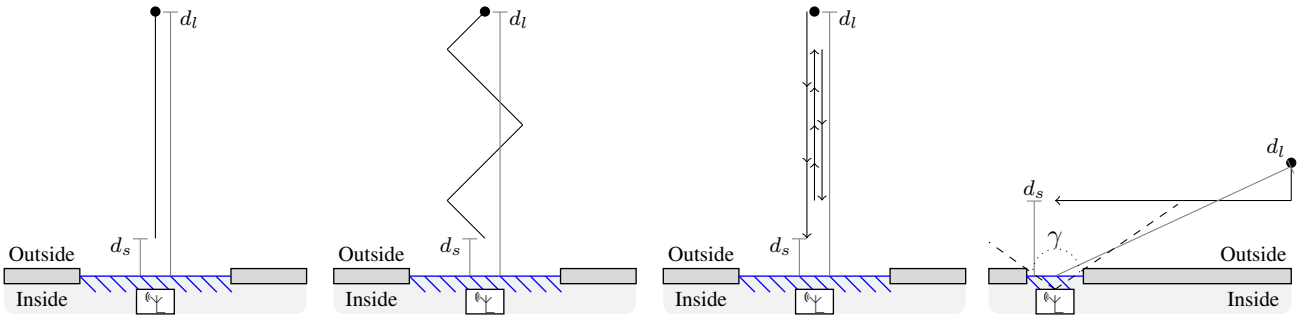



Fig. 5: Simulation of a zig-zag approach with two different choices of movement test window $w_l$. One is optimised for the actual speed, the other one for the effective velocity. The latter allows a faster detection.

In contrast to the direct approach, flying towards the window using a zig-zag approach results in a lower effective velocity towards the window, which in turn affects the choice of appropriate $w_l$ for the movement test. We simulated the standard deviation for a zig-zag approach with two different sliding windows; one for the actual speed $v$ and another for the effective velocity $v_{eff}$. In Figure 5, one can see that the detection with a window size that is too short for the effective velocity causes a delayed detection. Additionally, the shorter window size captures the zig-zag motion of the approach, whereas these erratic changes get smoothed out by the larger window size of $v_{eff}$. A similar effect is caused by a back-and-forth approach. The choice of $w_s$ for the free-space propagation test, on the other hand, is unaffected by the approach pattern.

If most of the approach is done out of LOS of the receiver, our assumptions about free-space propagation do not hold for the larger part of the approach. Therefore, the part of the approach that can be used for detection with our method gets reduced significantly. This distance depends on the surveillance distance $d_s$ and the FOV $\gamma$ of the window. But the detection might not always be possible. Depending on the FOV $\gamma$, noise threshold $\sigma$ and surveillance distance $d_s$, the approach in LOS can be too short to result in a notable change of the signal strength. Refer to Figure 6 to see how the detectability of a NLOS approach depends on $\gamma$ and $d_s$.




(a) Straight approach (b) Zig-zag approach (c) Back-and-forth approach (d) NLOS approach, with FOV $\gamma$

Fig. 4: Various example approach patterns

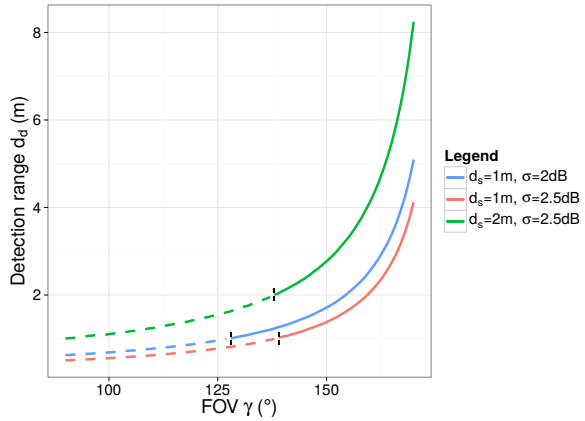



Fig. 6: Detection ranges for the NLOS approach. If the detection range $d_d$ is smaller than surveillance distance $d_s$, the detection will fail.

**Algorithm 1:** Detection algorithm

**Data:** Data flow of one transmitter
**Result:** Attack phase S
**begin**
  $S \longleftarrow \emptyset$
  **for** *Increasing* $w_l$ *until* $v_{\max}$ **do**
    **if** $s(w_s) < \sqrt{2}\sigma$ *and* $s(w_l) > \sqrt{2}\sigma$) **then**
      **if** $phase(w_l) > 0$ **then**
        $S \longleftarrow$ **Approach**
        **if** $proximity(w_l)$ **then**
          $S \longleftarrow$ **Surveillance**
          $reset(w_l)$;
      **else if** $phase(w_l) > 0$ **then**
        $S \longleftarrow$ **Escape**

### D. Detection algorithm

In order to detect drones, we compute the sliding windows for the standard deviation of the signal strength using $w_s$ and increasing values of $w_l$. We proceed like this until we find a value for which the standard deviation is below the threshold within $w_s$ but breaks the threshold for $w_l$.

We then use $w_l$ to determine the attack phase by computing $\Delta x$. For values greater than zero the drone is approaching. If $\Delta x$ additionally exceeds $\sigma_p$, then the drone has reached $d_s$ and an alarm is raised. The rolling windows should be reset after a proximity alert to make sure that the surveillance phase is properly detected. The surveillance alarm lasts until the start of the escape phase is detected.

Pseudo-code for the detection algorithm can be found in Algorithm 1, supporting the overall flow diagram in Figure 2. See Figure 7 for the movement and free-space propagation tests of the detection algorithm. This figure shows the rolling standard deviation for a moving transmitter and a receiver that is exposed to Gaussian noise. The attack phase tracking and proximity alert for the same simulation are pictured in Figure 8.

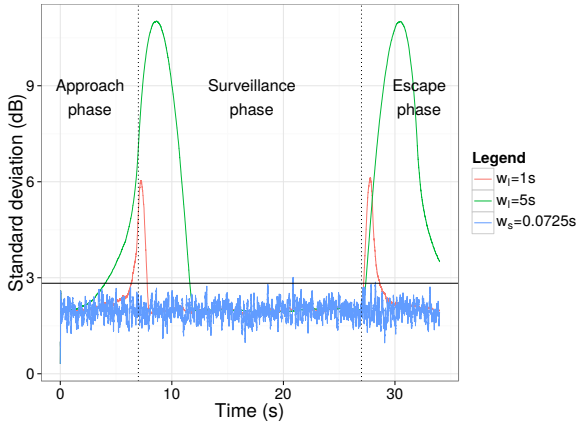



Fig. 7: Detection algorithm for a simulation with $v = 7m/s$, $d_l = 50m$ and Gaussian noise with $\sigma = 2dB$. It computes the standard deviation for the *free-space propagation* test during $w_s$ and several window sizes $w_l$ in parallel for the *movement* test. The black horizontal line is the noise threshold. It is first broken by $w_l = 5s$, making this is preferred window size.

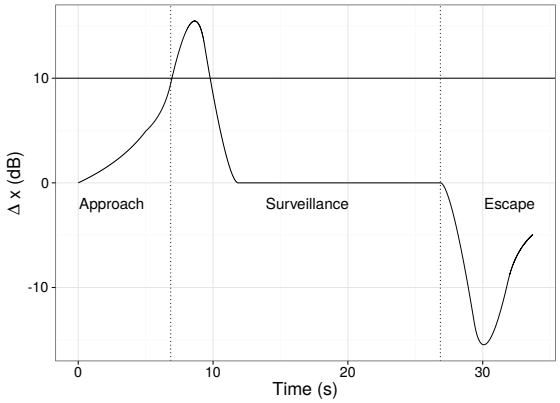



Fig. 8: Attack phase and proximity detection for a simulation with $v = 7$m/s, $d_l = 50$m, Gaussian noise with $\sigma = 2$dB and $\sigma_p = 10$. The black horizontal line is the proximity threshold. We used $w_l = 5$s as determined in Figure 7.

## VI. Experimental design

To demonstrate construction of our detection system using COTS hardware, we made use of a number of Raspberry Pi Model A[15] units as receivers. As this model lacks on-board Wi-Fi hardware, a Wi-Pi USB Wi-Fi adaptor was employed to receive packets. The Raspberry Pis ran the dumpcap utility [16] to capture Wi-Fi traffic, with the Wi-Pi adaptors in monitor mode to allow traffic capture without having to associate to any network.

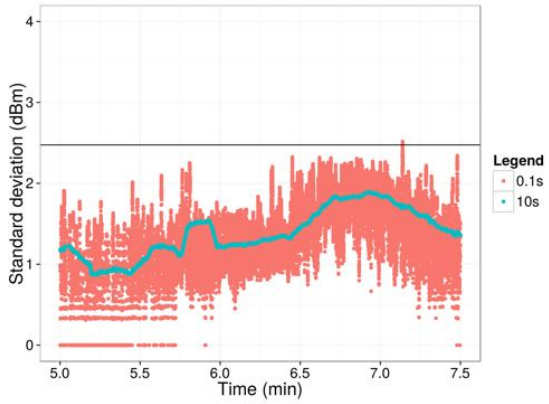
### A. Preliminary experiments

We conducted a preliminary experiment to investigate our assumptions about the signal properties of a flying drone, using a 3D Robotics X8+ drone. This drone was not equipped with a camera and thus enabled us to fly it in more-populated environments without breaching CAA regulations. Instead, the drone carried a Raspberry Pi that communicated with a laptop using the iPerf 2[17] benchmark tool at a bandwidth consistent with live video (4 MB/s, 400 packets/s). Calibration for the noise threshold was performed using background traffic from other Wi-Fi transmitters in the area.

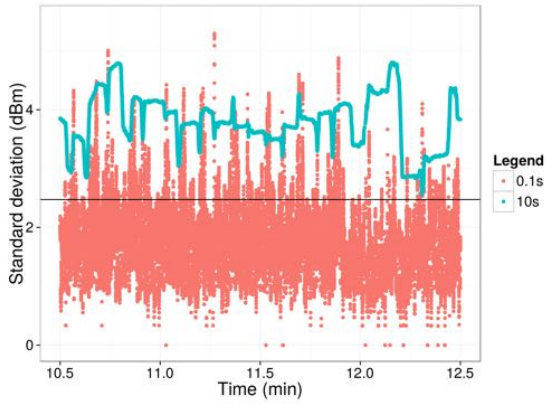
Over two short approaches (in distance and time), the expected results appeared; with the standard deviation staying above the threshold in the longer window during the approach and below the threshold in the shorter window. However, the receiver was configured to hop between the eleven 2.4 GHz Wi-Fi channels, resulting in a measured packet rate for the drone signal of only between 50 and 250 packets per second. As a result we refrained from using channel hopping in the remainder of this work, noting that using a number of network interfaces in parallel would alleviate the problem at only minimally-increased cost.

We also captured comparative data from stationary transmitters in LOS conditions in both static and dynamic environments. Once again we used Raspberry Pis as sender and receiver, executing iPerf 2 with the same parameters as in the drone experiment before. As expected, Figure 9 shows that a transmitter in a static environment does not breach the threshold, whereas the same transmitter violates the threshold even for the shorter window of the free-space propagation test. Such a transmitter is therefore not expected to be falsely detected by our system.

In an additional experiment, we gathered data of a different type of moving transmitter in the form of a Raspberry Pi being carried around. We captured transmissions both from moving indoors and outside in front of the window. In both cases, we moved within 10m of the receiver. The results of this experiment can be seen in Figure 10. Indoors, there is a high short-term standard deviation due to multipath effects. Even in the case of the movement outside an increased standard deviation of the longer window is accompanied with higher values breaching the threshold for the shorter window as well.



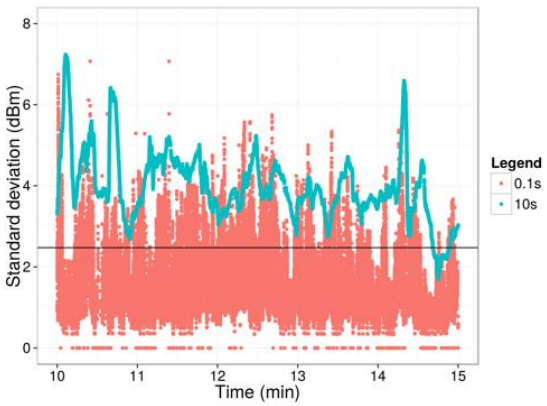

(a) Standard deviation of the signal strength in a static environment.



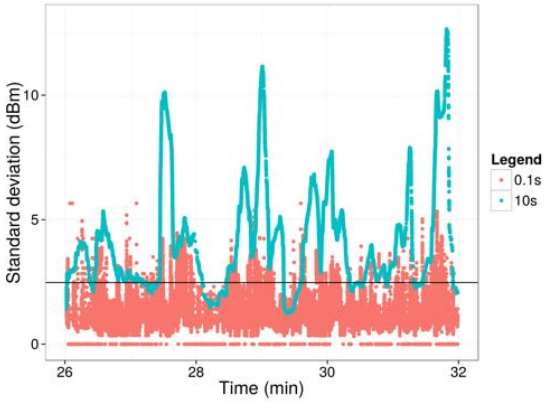

(b) Standard deviation of the signal strength in a dynamic environment.

Fig. 9: Results of the experiment with a static transmitter in LOS of the receiver.

[15] www.raspberrypi.org
[16] Part of the Wireshark network protocol analyser (www.wireshark.org)
[17] https://iperf.fr/




(a) Standard deviation of the signal strength indoors.




(b) Standard deviation of the signal strength outdoors.

Fig. 10: Results of the experiment with a moving transmitter in LOS of the receiver.

### B. Main drone experiments

The real-world experiments took place at a secluded[18] domestic property in Devon, United Kingdom. The property was an old farmhouse with thick, stone walls that were expected to attenuate Wi-Fi signals heavily. The property was surrounded by open land, allowing a variety of approach flight paths.

Two Raspberry Pis were installed in windows at the back side of the house; one in a first-floor bedroom and the other on the ground floor, in a kitchen. Figure 11 shows the deployment locations and one receiver in situ. The Raspberry Pis were connected to a controlling laptop via a Powerline network to avoid introducing cross-traffic on the wireless channel. For calibration, we used the beacons of an accesspoint that was located in the same room as the ground-floor receiver to gather enough calibration packets in a timeframe of thirty seconds. The signal strength measurements of these packets were then used to compute the standard deviation $\sigma$ of the baseline noise, as described in Section V-B.

[18]To comply with CAA regulations as detailed in Section II and avoid disruption to others. The nearest property that was not under our control was over 350m away.


(a) Outside view (b) Inside view

Fig. 11: Deployment of the receivers

Two very popular drone consumer drone models were used to mount our simulated privacy attacks. The primary candidate was a DJI Phantom 3 Standard, with the experiments being repeated using a Parrot Bebop unit for comparison. The two drone models are pictured in Figure 12. Unfortunately, some of the experiments were cut short due to time restrictions and the English weather; especially those from the second series using the Parrot Bebop.

(a) DJI Phantom 3 Standard (b) Parrot Bebop

Fig. 12: Drones used in the real-world experiments

We performed attack runs for each of the approaches described in Section V; in each case using the first floor window with Raspberry Pi 1 as the privacy-invasion target. The NLOS approach was further split into approaches over the roof and around the house to include both a descent from above and reaching the window from the side. Every approach in the series was performed with three run repetitions. The launch distance was between 55m and 65m for the LOS approaches. For the NLOS approaches the launch distance was much shorter at around 30m as they started from the front side of the house, with the operator moving to keep the drone in view during the approach. No run exceeded a top speed of 7m/s (approx. 25kph/15mph), as it is hard to control the drone at high speeds close to a building. The GPS flight data for each series were recorded by the Phantom. Some example traces are shown in Google Earth[19] plots in Figure 13. One can see that the GPS data became less precise close the window, as the drone no longer had a clear view of the GPS satellites. Specific details of the Phantom experiment series, as extract from the GPS flight data, can be found in Table IV.

[19]earth.google.com

(a) Straight approach (b) Zig-zag approach (c) NLOS approach over the roof (d) NLOS approach around the house

Fig. 13: Different flight patterns during the approach

| Pattern | Run | Distance (m) | Max. Speed (m/s) |
|---|---|---|---|
| Straight | 1 | 57 | 5.94 |
| | 2 | 61 | 6.24 |
| | 3 | 61 | 6.35 |
| Zig-zag | 1 | 61 | 6.36 |
| | 2 | 62 | 6.66 |
| | 3 | 62 | 4.72 |
| Back-and-forth | 1 | 62 | 4.24 |
| | 2 | 61 | 4.43 |
| | 3 | 61 | 5.38 |
| NLOS over the roof | 1 | 28 | 2.78 |
| | 2 | 30 | 3.92 |
| | 3 | 30 | 3.62 |
| NLOS around the house | 1 | 35 | 2.48 |
| | 2 | 31 | 3.48 |
| | 3 | 33 | 4.06 |

TABLE IV: DJI Phantom 3 experiment series.

| Parameter | Value | Description |
|---|---|---|
| $d_s$ | 1m | We stayed within one to two meters of the window during the surveillance phase |
| $d_l$ | 50m | All the LOS approaches started further than that |
| $w_s$ | 0.1s | Corresponding to a maximum velocity of 10 m/s |
| $w_l$ | 5s, 10s, 15s, 30s | Corresponding to the following respective minimal velocities: 5.075, 2.535, 1.69, 0.845 m/s |
| $\sigma$ | 1.75 | Derived according to Section V-B; hence the noise threshold is $\sqrt{2} \cdot 1.75$ |
| $\sigma_p$ | 10 | The threshold for the proximity alert; derived from $d_s$ |

TABLE V: System parameter selections used in Evaluation

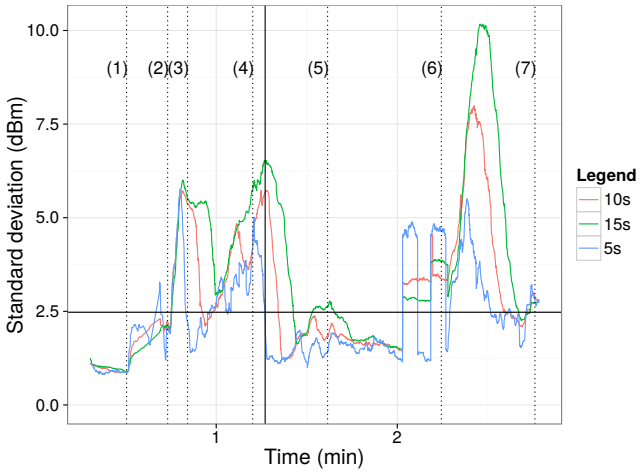
## VII. Evaluation

In this section, evaluate how well the metrics from Section V are able to detect a drone using the data from our experiments. We use only the data from the Phantom for the analysis of the movement test, attack-phase tracking and proximity detection. The results for these are similar for both Phantom and Bebop and, as described in Section VI, we have a more complete set of runs for the Phantom. However, we realised when analysing the data that we only captured the control channel of the Phantom. Therefore, the packet rate is quite low at 40 packets per second. While this is not a problem for the large-window tests, it only leaves us with about four packets within $w_s$. Hence, we used the data from the Bebop series to examine the free-space propagation test.

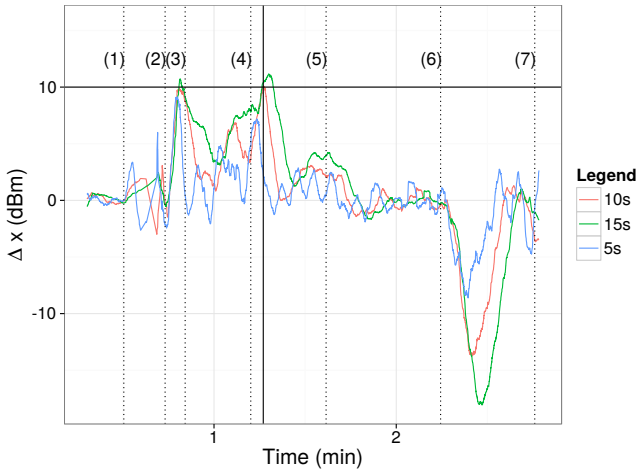
In our evaluation we selected system parameters as given in Table V. The minimal detection range using these parameters was 24.5m.

### A. Straight approach

Examining movement-tracking first, in Figure 14a three windows of rolling standard deviation are displayed for the first run of the Straight Phantom set. The drone launched at (1) at a distance of 57m from the window; the first increase is due to the initial ascent, but is not enough to break the threshold. There is a short spike in the 5s window before the actual detection happens which only lasts for a few samples. At (2) the drone started accelerating towards the window. Due to its slow speed at the time, the 10s window is the first to detect it, albeit by a very small margin. The detection happens surprisingly fast after only one and half a meters travelled since takeoff. From the model in Section V we had expected the drone to be on target height and approach speed when we received our first packet, however the detection happened much earlier than expected because the takeoff and the varying speed due to acceleration increase the measured standard deviation. In comparison, if we were to start detection during the second acceleration period starting at (3) at a distance of 51m, the drone would be detected 37m from the window. This is closer to the expected value of 25.15m for this distance but still earlier than predicted. This can be attributed to the drone not travelling as close to a constant speed as expected. The velocity decreased at (4), as we had to slow down to pilot the drone safely closer to the house; at distance 6m. The larger window size of 15s compensates for decelerating during the approach and throughout this period the standard deviation stays above the threshold (recall that our system always chooses the lowest window size that is *above the threshold* and uses it for phase tracking and proximity detection). The surveillance period started at (5) where the drone stayed as still as possible in front of the window. After the rolling windows catch up, the standard deviation drops markedly and stays below the threshold almost the entire time until the escape phase starts at (6). However, it does still breach the threshold occasionally. In practice there are still many corrections necessary to keep the drone in place in front of the window. As the drone is so close to the receiver during the surveillance phase, even small movements have a notable effect on the RSS and can breach the detection threshold. Finally, the drone landed at (7) leading to another short spike as it reached the ground.




(a) Standard deviation of the signal strength. The black horizontal line is the movement detection threshold.




(b) Attack phases. The drone is approaching when the values are positive and escaping for negative values. The black horizontal line is the proximity threshold.

Fig. 14: Straight approach with the DJI Phantom 3. The black vertical line shows when the 5s window undercuts the movement threshold. The dotted vertical lines indicate: (1) Takeoff, (5) Surveillance, (6) Escape and (7) Landing. More details can be found in the text.

For attack-phase tracking, we compare the first and second halves of $w_l$, shown in Figure 14b. Both the 10s and 15s windows stay positive for the entire approach. There is more variation in the 5s window as it captures slowing down better than the longer window sizes. All the windows have values close to zero during surveillance after catching up and are negative during the escape phase, as per our expectations.

A proximity alert is triggered when the phase tracking breaches threshold $\sigma_p$. In Figure 14b, one can see that with the larger window size of 15s, readings break the threshold early due to the cumulative effect of positive readings over the preceding period. However, at this moment the 15s window is not the smallest for which the standard deviation is above the

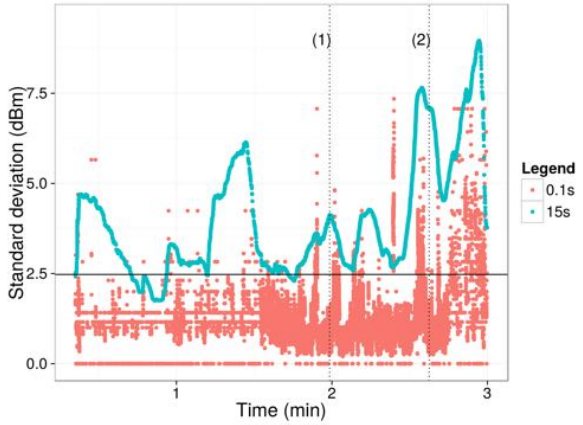


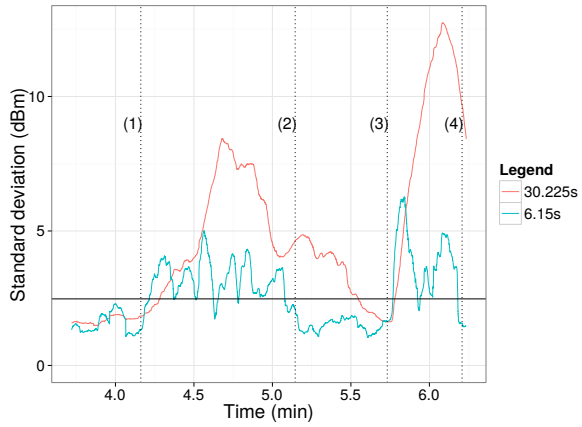

Fig. 15: Standard deviation for a straight approach with the Parrot Bebop. The black horizontal line is the free-space propagation detection threshold which should not be exceeded within $w_s = 0.1s$. The vertical lines indicate: (1) Surveillance and (2) Escape. More details can be found in the text.

threshold and therefore another window size is favoured over it for phase tracking and the proximity metric — so no spurious proximity warning is produced. The second peak, just after (4), results in a genuine proximity alert for the 10s window. The distance from the window at the time was 4.2m, which is greater than $d_s$. In Figure 14a, the black vertical line shows the moment when values in the 5s window fall below the detection threshold. Consequently, the 10s window is used to check for proximity and the alarm is triggered.

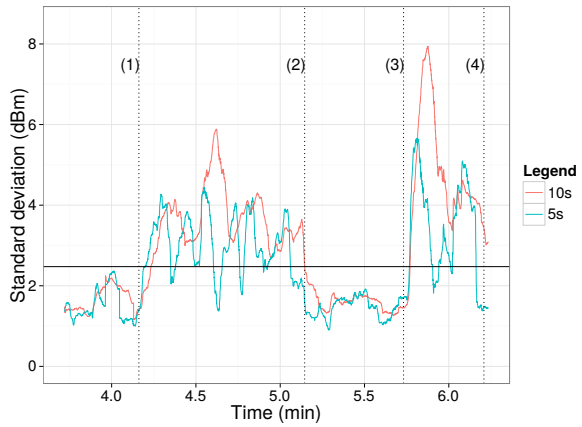
As mentioned before, we evaluate the free-space propagation test that uses $w_s$ using the Bebop series due to the higher observed packet rate. Figure 15 shows the RSS standard deviation for the first straight Bebop run. The standard deviation of the 0.1s window stays mostly below the threshold throughout the approach; only very few samples violate it initially. When the drone is close to the wall of the building, the standard deviation increases even within $w_s$, hence breaking the threshold more often. At this time, however, we can already be quite sure that it is a drone. The peaks during the surveillance period are caused by movement in front of the window; we had to make more corrections to account for drift with the Bebop, than with the heavier Phantom.

### B. Zig-zag approach

For the zig-zag approach we will first look at window sizes derived from the maximal and effective approach velocity. For the first zig-zag run the maximal approach velocity was $v_{\max} = 5.12$m/s. Using $v_{\max}$ we compute $w_{l,\max} = 6.15$s. On the other hand, the effective velocity for the whole approach was only $v_{eff} = 1.04$m/s and thus $w_{l,eff} = 30.225$s. In Figure 16a one can see that the smaller window size detects the drone faster at a distance of 59.5m whereas the larger window size only detects the drone 53m away from the receiver. However, $w_l = 6.15$s captures the erratic changes caused by the zig-zag movement, whereas $w_l = 30.225$s smooths out the flight pattern and shows the whole approach. The decrease in standard deviation in the second part of the approach is caused by a general decrease in approach speed.




(a) Parameters derived from the model using $v_{\max}$ and $v_{eff}$




(b) System parameters

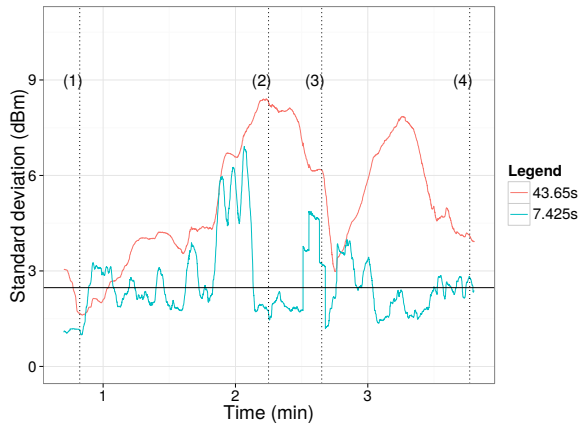
Fig. 16: Standard deviation for a zig-zag approach with the DJI Phantom 3. The black horizontal line is the movement detection threshold. The vertical lines indicate: (1) Takeoff, (2) Surveillance, (3) Escape and (4) Landing. More details can be found in the text.

We can see the same effect if we use our system with the parameters from the beginning of the section, cf. Figure 16b. Lower window sizes than the derived values are possible as $w_l = 10$s already captures the complete approach. This is again likely to be due to an increased standard deviation because of the launch and acceleration periods.

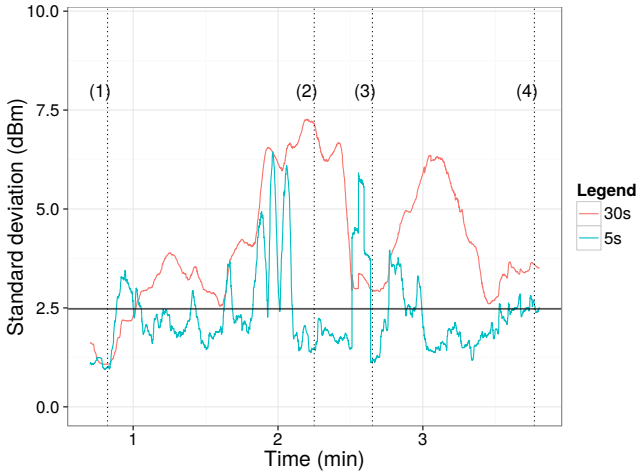
### C. Back-and-forth approach

For the back-and-forth approach, we proceed in a similar fashion as for the zig-zag approach. We compute $w_l$ for the maximal speed $v_{\max} = 4.24$s, and the effective velocity $v_{eff} = 0.72$s. Similarly to before, Figure 17a shows that the former is affected by the flight pattern whereas the latter captures the complete approach. The shorter window $w_{l,\max} = 7.425$s shows larger peaks followed by smaller peaks. The larger peaks correspond to forward, the lower peaks to backward motion.

In comparison, the higher window size $w_{l,eff} = 43.65$s is able to capture the complete approach as a whole. However, it also covers the entire surveillance period, again highlighting the need for the proximity metric as a stop-and-alarm condition. If we use the system parameters as before, we can see in Figure 17b that the 30s window is sufficiently large.




(a) Parameters derived from the model using $v_{\max}$ and $v_{eff}$




(b) System parameters

Fig. 17: Standard deviation for a back-and-forth approach with the DJI Phantom 3. The black horizontal line is the movement detection threshold. The vertical lines indicate: (1) Takeoff, (2) Surveillance, (3) Escape and (4) Landing. More details can be found in the text.

Additionally, we can study the approach phase tracking in Figure 18. We can see that if we only rely on the shorter window, the system oscillates between approach and escape phases, whereas the longer window size makes it possible for us to detect the approach as a larger trend.

### D. NLOS approach over the roof

The first NLOS run with an approach over the roof is pictured in Figure 19a. The drone takes off at launch distance 28m on the other side of the house. Its forward acceleration starts at (2) and it reaches maximal altitude at (3). At (4) the

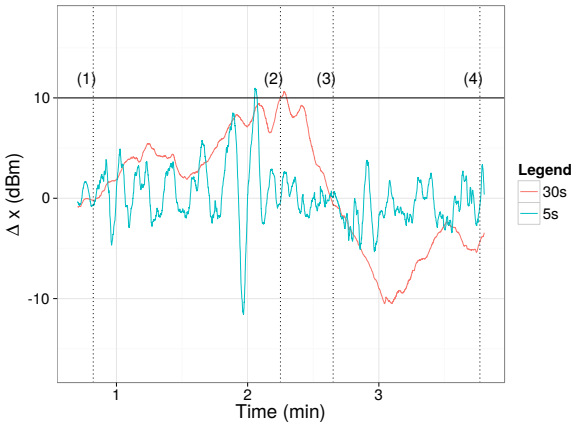



Fig. 18: Attack phases for a back-and-forth approach with the DJI Phantom 3. The black horizontal line is the proximity detection threshold. The drone is approaching when the values are positive and escaping for negative values. The vertical lines indicate: (1) Takeoff, (2) Surveillance, (3) Escape and (4) Landing. More details can be found in the text.

descent begins, on the receiver side of the house. The first big peak for window size 15s happens as the drone is flying over the receiver, above the roof. The next peak occurs as it is descending into LOS, with another one following when it gets closer to the window up to $d_s$. The surveillance period starts at (5), before the drone escapes over the roof again at (6).

We had to fly at slow speeds close to the house for careful manoeuvring which explains why higher values are needed to capture the whole approach compared to the straight LOS series.

In Figure 19b, the standard deviation within $w_s$ is displayed. One can see that the standard deviation is low in spite of the missing LOS connection. The roof of the house seems to affect the signal only by attenuation and does not lead to significant interference due to multipath effects.

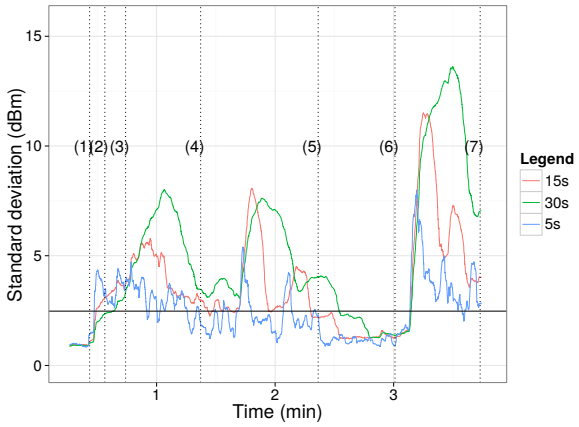
### *E. NLOS approach around the house*

A run of the other NLOS approach variant, around the house, is shown in Figure 20. Similarly to the approach over the house, we see peaks as the drone crosses the receiver, gets into LOS and approaches the window. For both NLOS approaches there is a distinct increase in the standard deviation when the drone comes into LOS. Once it is in LOS the behaviour is similar to the LOS approaches.

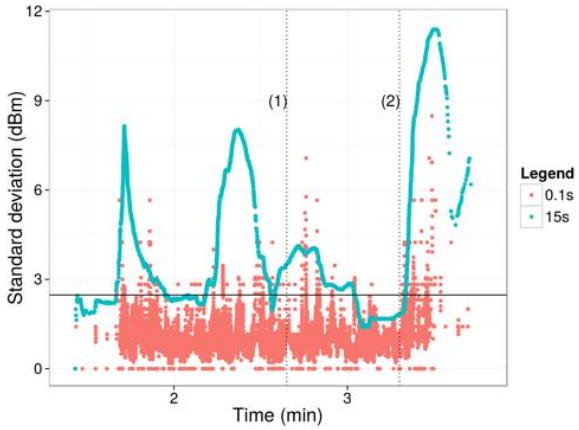
### *F. Detection ranges*

To examine detection range we let several rolling windows (5, 10, 15, 30 seconds) run in parallel and chose the first window size to detect the drone as well as the first that stayed above the threshold for the whole approach phase.

Table VI shows the resulting movement detection ranges. In all of the LOS runs, our system detected the drones far earlier than the minimal detection range of 24.5m, as measured from GPS traces using Google Earth. The detection distances of the second and third back-and-forth runs are lower than the




(a) Movement test with DJI Phantom 3. The vertical lines indicate: (1) Takeoff, (5) Surveillance, (6) Escape and (7) Landing. More details can be found in the text.




(b) Free-space propagation test with Parrot Bebop. The vertical lines indicate: (1) Surveillance and (2) Escape. More details can be found in the text.

Fig. 19: Standard deviation for a NLOS approach over the roof. The black horizontal line is the noise threshold.

| Pattern | Run | Detection range (m) |
|---|---|---|
| Straight | 1 | 55.5 |
| | 2 | 55 |
| | 3 | 53.77 |
| Zig-zag | 1 | 60 |
| | 2 | 59.5 |
| | 3 | 60.13 |
| Back-and-forth | 1 | 60.22 |
| | 2 | 51.47 |
| | 3 | 48.65 |

TABLE VI: Detection ranges of the DJI Phantom 3 LOS experiment series.

first as they began with a longer and faster forward movement and thus still resembled a straight approach at the time of detection. That the drone flew the approaches with varying velocity actually increased the detection range, as did the more time-consuming flight patterns.

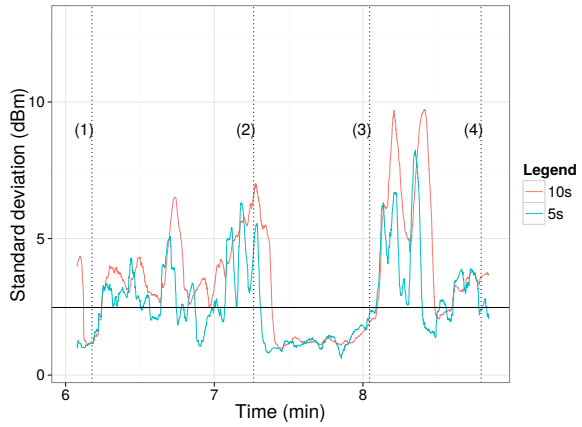



Fig. 20: Standard deviation for a NLOS approach around the house with the DJI Phantom 3. The black horizontal line is the movement detection threshold. The vertical lines indicate: (1) Takeoff, (2) Surveillance, (3) Escape and (4) Landing. More details can be found in the text.

The proximity alert was triggered within 5m of the target window in every experiment run. Unfortunately, due to the roof occluding GPS satellites, the data from the traces is too imprecise to determine the exact distance in each case.

## VIII. Discussion

### A. Results

The most notable discovery throughout our experiments was that detection happened earlier than our simulations had predicted. In large part this is simply attributable to our model being very conservative and real-world attacks not demonstrating such extreme behaviour. Ever-changing conditions and operator inaccuracy leads real drone flight to have far more speed variance than the constant-speed approaches we had assumed. No two consecutive flights were identical, even with the same intended approach pattern. Close to the target, in the latter stages of an approach and during surveillance, the movement is even more erratic as the drone must navigate a tighter environment and counteract drift from wind conditions and sidewash turbulence in order to establish and maintain a close hover. The reliance purely upon the operator's precise control is intensified as GPS stabilisation systems are hampered by a reduced view of the sky when near a building. The detection system parameters must be carefully selected to ensure that the Escape phase is not indicated prematurely.

Takeoff is the first movement a drone makes, but communication begins before this point. This prior communication is beneficial to our detection as the received messages establish an initial measurement in each window, such that the takeoff itself is enough to bring the standard deviation above the detection threshold in many cases. Such early detection gives ample time for the user to be warned of a privacy-invasion attack and take protective action. It also presents an opportunity to simply inform a user of nearby drone activity that is not, or has not yet developed into, an attempt at privacy invasion.

### B. Observations

Our experimental experiences reinforced the expectation of a reliance by the pilot on the drone's FPV video stream. At close range it is possible (and sometimes preferable) to watch the drone directly when flying, however the benefits of doing this quickly disappear as the drone operates further away and the pilot is less able to reason about its position relative to other objects purely by eye. In this case, and certainly where there is no direct line of sight at all, the attacker depends on the streamed video to pilot successfully. Even in the minimal case, conducting the privacy attack itself, the operator needs immediate feedback to ensure that the drone obtains a good view of the interior of the building — it proved easy to capture detailed footage of a window-frame by mistake. An attacker could, if their hardware permitted, disable the video stream for a period in order to avoid providing a suitable communication stream for the detection system. However, doing so near the target or when attempting surveillance, would likely jeopardise the attack. Indeed, a short surveillance distance is crucial in mounting a successful attack; at large distances a high-resolution camera or telescopic lens is required to capture a detailed image of the building interior and even then the visible area is heavily constrained by the window. Our observations suggest that values of $d_s$ at one or two metres are realistic.

If the drone flies close to the ground for the entirety of the approach, ground reflections are strong multipath components and violate the free-space propagation assumption; making the drone indistinguishable from an attacker moving on the ground themselves. In practice however, we expect that the attacker has to overcome access restrictions and hence must fly higher for at least some part of the approach, making the drone detectable again.

## IX. Future Work

Our experiments were conducted with the drone and controller as the only communicating parties on a Wi-Fi channel. In order to understand better the expected performance of the system in an urban environment, it would be helpful to examine the effect of cross-traffic on detection. As the cross-traffic packet-rate increases, the number of successfully-measured RSS samples in a given period is reduced and we would expect this to degrade the detection speed and accuracy, although the precise behaviour remains unclear.

As this detection approach uses only RSS measurements, the equipment used in this work is but one of a great number of possibilities. The majority of Wi-Fi hardware provides RSS data and many drivers make this information available. Purely software-based implementations are feasible for manufacturers and often for end-users as well. For example, an Android user-space driver exists for some Wi-Fi adapters[20], allowing the system to be implemented as an app to turn a user's mobile device into a drone detector. The low modification requirements of this approach greatly benefit its applicability, especially if they mean it can be incorporated into commonplace devices instead of requiring additional hardware. This does raise the question however, of how well the system would perform with an unordered, potentially dynamic, deployment. Understanding

[20]For example, the Alfa One NIC, with an RTL 8187 chipset

the factors involved would greatly inform discussions of the widespread applicability of the system.

## X. Conclusion

In this work, we developed a method to detect privacy-invasion attacks by drones based on their communication with a controller. Our approach uses analysis of RSS variance in the drone's transmissions to check for free-space propagation and examine its movement. We developed a further method to monitor if the drone is approaching the detector or moving away from it and another to detect proximity to a receiver. Combined use of these metrics enables us to track the phase of an attack. We conducted real world experiments using two types of commercially-available drones that communicate over Wi-Fi, with various example flight patterns and target windows. For all series, the approaching drone was detected early on during the attack and the system successfully triggered an alarm when the drone got close to the window. We were able to detect a drone approaching in LOS of the detection system at a minimal distance of 48 meters. Even for a NLOS approach, detection was fast and actually happened before the drone came into LOS.

In summary, our system is able to detect drones flying nearby and can alert when a drone is in proximity of a window. For even the largest of physical windows, the detection happens at a great enough range that an attacker will have had no chance to conduct detailed surveillance before the alarm is raised. Our implementation used only cheap and easily-obtained hardware. Moreover, the system is built upon measurements that are available in the vast majority of Wi-Fi capable systems, opening up widespread deployment options.

## Acknowledgment

Simon Birnbach and Richard Baker are supported by the EPSRC.

## References

[1] Guard From Above. Intercepting hostile drones, 2016. Accessed: 10.07.2016. URL: http://guardfromabove.com/.

[2] Federal Aviation Agency. FAA releases 2016 to 2036 Aerospace Forecast, 2016. Accessed: 29.06.2016. URL: http://www.faa.gov/news/updates/?newsId=85227.

[3] Federal Aviation Agency. FAA releases drone registration location data, 2016. Accessed: 29.06.2016. URL: https://www.faa.gov/news/updates/?newsId=85548.

[4] Civil Aviation Authority. The Air Navigation Order 2016 And Regulations (CAP393), 2016. Accessed: 25.08.2016. URL: http://publicapps.caa.co.uk/docs/33/CAP%20393_AUG2016.pdf.

[5] Sanjay K Boddhu, Matt McCartney, Oliver Ceccopieri, and Robert L Williams. A collaborative smartphone sensing platform for detecting and tracking hostile drones. In *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2013.

[6] Joël Busset, Florian Perrodin, Peter Wellig, Beat Ott, Kurt Heutschi, Torben Rühl, and Thomas Nussbaumer. Detection and tracking of drones using advanced acoustic cameras. In *SPIE Security+ Defence*. International Society for Optics and Photonics, 2015.

[7] Ellen E Case, Anne M Zelnio, and Brian D Rigling. Low-cost acoustic array for small UAV detection and tracking. In *2008 IEEE National Aerospace and Electronics Conference*. IEEE, 2008.

[8] British Broadcasting Corporation. Big rise in drone jail smuggling incidents, 2016. Accessed: 29.06.2016. URL: http://www.bbc.co.uk/news/uk-35641453.

[9] Kristina Davis. Two plead guilty in border drug smuggling by drone. Los Angeles Times, 2015. Accessed: 29.06.2016. URL: http://www.latimes.com/local/california/la-me-drone-drugs-20150813-story.html.

[10] Drone Detector. Compare detection systems, 2016. Accessed: 10.07.2016. URL: http://www.dronedetector.com/compare-detection-systems/.

[11] T Eshel. Mobile radar optimized to detect UAVs, precision guided weapons. *Defense Update*, 2013.

[12] Vijay Garg. *Wireless communications & networking*. Morgan Kaufmann, 2010.

[13] Sait Murat Giray. Anatomy of unmanned aerial vehicle hijacking with signal spoofing. In *Recent Advances in Space Technologies (RAST), 2013 6th International Conference on*, pages 795–800. IEEE, 2013.

[14] NPD Group. Year-over-year drone revenue soars, according to NPD, 2016. Accessed: 29.06.2016. URL: https://www.npd.com/wps/portal/npd/us/news/press-releases/2016/year-over-year-drone-revenue-soars-according-to-npd/.

[15] Samy Kamkar. Skyjack, 2013. Accessed: 10.07.2016. URL: http://samy.pl/skyjack/.

[16] Nitis Mukhopadhyay. *Probability and statistical inference*. CRC Press, 2000.

[17] Kavitha Muthukrishnan, Berend Jan van der Zwaag, and Paul Havinga. Inferring motion and location using WLAN RSSI. In *Mobile Entity Localization and Tracking in GPS-less Environnments*, pages 163–182. Springer, 2009.

[18] Matthew Peacock and Michael N Johnstone. Towards detection and control of civilian unmanned aerial vehicles. 2013.

[19] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Flying objects detection from a single moving camera. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[20] Stephan Sand, Armin Dammann, and Christian Mensing. *Positioning in Wireless Communications Systems*. John Wiley & Sons, 2014.

[21] Michael S. Schmidt. Secret service arrests man after drone flies near White House. New York Times, 2015. Accessed: 29.06.2016. URL: http://www.nytimes.com/2015/05/15/us/white-house-drone-secret-service.html.

[22] Michael S. Schmidt and Michael D. Shear. A drone, too small for radar to detect, rattles the White House. New York Times, 2015. Accessed: 29.06.2016. URL: http://www.nytimes.com/2015/01/27/us/white-house-drone.html.

[23] Linzi Sheldon. Woman terrified by drone outside her window. KIRO 7, 2014. Accessed: 29.06.2016. URL: http://www.kiro7.com/news/woman-terrified-drone-outside-her-window/81721261.

[24] Daniel P Shepard, Jahshan A Bhatti, Todd E Humphreys, and Aaron A Fansler. Evaluation of smart grid and civilian UAV vulnerability to GPS spoofing attacks. In *Proceedings of the ION GNSS Meeting*, volume 3, 2012.

[25] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. Rocking drones with intentional sound noise on gyroscopic sensors. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 881–896, 2015.

[26] Michigan Tech. Robotic falconry, 2016. Accessed: 10.07.2016. URL: http://me.sites.mtu.edu/rastgaar/home/news/.

[27] Robotics Trends. Drone registration no drag as sales soar, 2016. Accessed: 29.06.2016. URL: http://www.roboticstrends.com/article/drone_registration_no_drag_as_sales_soar.

[28] Juan R Vasquez, Kyle M Tarplee, Ellen E Case, Anne M Zelnio, and Brian D Rigling. Multisensor 3D tracking for counter small unmanned air vehicles (CSUAV). In *SPIE Defense and Security Symposium*. International Society for Optics and Photonics, 2008.

[29] WDRB. Hillview man arrested for shooting down drone; cites right to privacy, 2015. Accessed: 29.06.2016. URL: http://www.wdrb.com/story/29650818/hillview-man-arrested-for-shooting-down-drone-cites-right-to-privacy.

[30] Zhuoling Xiao, Hongkai Wen, Andrew Markham, Niki Trigoni, Phil Blunsom, and Jeff Frolik. Identification and mitigation of non-line-of-sight conditions using received signal strength. In *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 667–674. IEEE, 2013.