GyrosFinger: Fingerprinting Drones for Location Tracking Based on the Outputs of MEMS Gyroscopes

YUNMOK SON, JUHWAN NOH, JAEYEONG CHOI, and YONGDAE KIM, KAIST, Republic of Korea

Drones are widely used for various purposes such as delivery, aerial photography, and surveillance. Considering the increasing drone-related services, tracking the locations of drones can cause security threats such as escaping from drone surveillance, disturbing drone-related services, and capturing drones. For wirelessly monitoring the status of drones, telemetry is used, and this status information contains various data such as latitude and longitude, calibrated sensor outputs, and sensor offsets. Because most of the telemetry implementation supports neither authentication nor encryption, an attacker can obtain the status information of the drones by using an appropriate wireless communication device such as software-defined radio. While the attacker knows the locations of the drones from the status information, this information is not sufficient for tracking drones because the status information does not include any identity information that can bind the identity of the drone with its location.

In this article, we propose a fingerprinting method for drones in motion for the binding of the identity of the drone with its location. Our fingerprinting method is based on the sensor outputs included in the status information, i.e., the offsets of micro-electro mechanical systems (MEMS) gyroscope, an essential sensor for maintaining the attitude of drones. We found that the offsets of MEMS gyroscopes are different from each other because of manufacturing mismatches, and the offsets of five drones obtained through their telemetry are distinguishable and constant during their flights. To evaluate the performance of our fingerprinting method on a larger scale, we collected the offsets from 70 stand-alone MEMS gyroscopes to generate fingerprints. Our experimental results show that, when using the offsets of three and two axes calculated from 128 samples of the raw outputs per axis as fingerprints, the F-scores of the proposed method reach 98.78% and 94.47%, respectively. The offsets collected after a month are also fingerprinted with F-scores of 96.58% and 78.45% under the same condition, respectively. The proposed fingerprinting method is effective, robust, and persistent. Additionally, unless the MEMS gyroscope is not replaced, our fingerprinting method can be used for drone tracking even when the target drones are flying.

CCS Concepts: • **Security and privacy → Systems security**; **Embedded systems security**; • **Computer systems organization → Embedded systems**; **Sensors and actuators**;

Additional Key Words and Phrases: Device fingerprinting, MEMS gyroscope, sensor, security

ACM Reference format:
Yunmok Son, Juhwan Noh, Jaeyeong Choi, and Yongdae Kim. 2018. GyrosFinger: Fingerprinting Drones for Location Tracking Based on the Outputs of MEMS Gyroscopes. *ACM Trans. Priv. Secur.* 21, 2, Article 10 (February 2018), 25 pages.
https://doi.org/10.1145/3177751

This work was supported by an Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (Grant No. B0717-16-0119, Development of Information Leakage Prevention and ID Management for Secure Drone Services).

Authors' addresses: Y. Son, J. Noh, J. Choi, and Y. Kim, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon, 34141, Republic of Korea; emails: (yunmok00, juwhan, go1736, yongdaek)@kaist.ac.kr.

https://doi.org/10.1145/3177751

1 INTRODUCTION

A drone is a type of unmanned aircraft system (UAS), whose components include a flight controller, an inertial measurement unit (IMU), multiple rotors, and a battery. At present, drones are widely used for distribution delivery, aerial photography, and surveillance, as well as for private hobbies. Because of the increasing drone population, tracking drones has become an important issue for both improving their safety and attacking drones.

Tracking drones can be used to enhance their safety, as it can be used to avoid collisions, improve traffic efficiency, and prevent the flights of unauthorized drones, considering the increasingly crowded airspace. For example, the National Aeronautics and Space Administration (NASA) and the Federal Aviation Administration (FAA), in cooperation with various companies, such as Amazon and Google, have been developing the UAS traffic management (UTM) system for drones flying at low altitudes between 200 and 500 feet. In the UTM system, location tracking for the drones is necessary to manage the drone traffic using additional means (such as an automatic dependent surveillance-broadcast (ADS-B) or "ADS-B-like" system) that transmit the identity and location of the drone to the UTM system [50, 53, 54]. On the other hand, tracking drones can be used for malicious purposes as well. An attacker can maliciously track the locations of the drones for escaping from drone surveillance, disturbing drone-related services, and capturing drones themselves. In this article, from adversarial viewpoints, we introduce a fingerprinting method for drones that can be used to track their locations based on the outputs of an essential sensor of the drones even during flight. Most drones have a telemetry functionality based on the micro aerial vehicle link (MAVLink) protocol [28] for monitoring their status. Because the drones are used for flying, the status information supported by this telemetry is usually given via wireless communication modules, such as 433 or 915MHz telemetry and Wi-Fi transceivers, in real time. Transmitted status data are different from each implementation of flight controllers, but most implementations include system parameters, global positioning system (GPS) data, altitude, rotor status, memory status, sensor status, and sensor outputs. These data are used not only to monitor but also to debug the drone systems. An attacker can collect the data using an appropriate radio device because many telemetry modules have neither authentication nor encryption [11, 23, 27, 41, 43]. Thus, an attacker can find the locations of drones from the GPS data as well as altitude included in the status information, but cannot track each drone, because there is no identifier that can be used for binding drones and the location information. In other words, essential information needed to track drones is a *persistent identifier* that does not change before, during, and after the flight.

For this purpose, we propose a fingerprinting method that can identify drones based on the offset information of microelectromechanical system (MEMS) gyroscopes readily available from the status information. MEMS technology has been used to implement extremely small mechanical structures in chips, and the MEMS gyroscope measures angular speed by sensing the force caused by rotation in the MEMS structure. Because of mismatches in manufacturing processes, every MEMS gyroscope is slightly different from the others. This leads to minute differences in the outputs of each MEMS gyroscope even for identical inputs, and these differences can be used as fingerprints. These differences are represented by the offsets of MEMS gyroscopes for three axes (*X*, *Y*, and *Z*), which are measured by averaging the raw outputs of MEMS gyroscopes in a stationary state for calibrating the gyroscopes. The values of the offsets are determined when the drones are turned on (for calibration in a stationary state): these offsets do not change during flight.

To evaluate our method, we collected fingerprints from five commercial drones through telemetry before, during, and after the flight. We found that all fingerprints 1) are distinguishable and 2) do not change before, during, and after the flight. We further confirmed the latter by manually analyzing each drone's firmware. To confirm if offsets can be used as fingerprints, we ran

Table 1. Comparison with Existing Device Fingerprinting Methods Based on MEMS Sensors

Work	MEMS Sensor*	Condition	Target devices	Data for fingerprint	Fingerprinting method	Time for learning	Results
Bojinov et al. [4]	Acc	Lab & public (with flipping devices)	Smart devices with web browsers	Calibrated sensor outputs (using JavaScript)	Weighted Euclidean distance based on sensitivity and offset	–	100% accuracy (Lab), 15.1% accuracy (Public)
Accel-Print [10]	Acc	Lab (stationary, motor vibration)	Smart devices with web browsers, sensors	Calibrated sensor outputs (using SPI & JavaScript)	Machine learning based on 36 features among 80 extracted features	30 seconds	99% accuracy, 96% precision & recall
Das et al. [9]	Acc & Gyro	Lab & public (stationary, in hand, audio)	Smart devices with web browsers	Calibrated sensor outputs (using JavaScript)	Machine learning based on 70 features among 100 extracted features	30–40 seconds	99% F-score (Lab, Acc & Gyro), 95% F-score (Lab, Gyro)
Gyros-Finger	Gyro	Lab (stationary, in motion)	Drones & sensors	Sensor offsets (using telemetry & SPI)	Bounded Euclidean distance based on offsets for each axis	1.28 seconds†	98.78% F-score (Using 3 axes), 94.47% F-score (Using 2 axes)

*Acc: accelerometer, Gyro: gyroscope

†If the offsets already calculated can be obtained from the target devices (i.e., through the telemetry of drones), this time is not required.

large-scale experiments as well. We collected over 40 million samples of raw data in a stationary environment from 70 MEMS gyroscope chips (five models from two manufacturers) for each axis with a sampling rate of 100Hz. Our experiment shows an F-score of 98.78% for generating fingerprints using the offsets of the three axes averaged over 128 samples (collected for about 1.28 seconds in our experiment), and an F-score of 94.47% on average, using the offsets of two axes under identical conditions. We collected additional data after about a month to verify whether our fingerprints are time-invariant. After a month, the F-scores decreased to 96.58% and 78.45% when using the offsets of three and two axes, respectively, because of the changes in offsets over time. As drones are using offsets of three axes, this shows that gyroscope offsets can be used as a persistent identifier for tracking drones. Because our fingerprint is based on hardware characteristics, it can be used without modifying the software and hardware of commercial drones. Moreover, our method can be applied to UTM for tracking commercial drones that are already released in the market without any modification and additional cost.

As a side note, our study related with the previous fingerprinting methods using MEMS sensors [4, 9, 10] to fingerprint smart devices with web browsers. While the purposes are different, the comparison of our work with the previous studies is summarized in Table 1. To the best of our knowledge, the proposed method is the first device-fingerprinting method that uses only MEMS gyroscopes[1] and targets embedded devices *without web browsers* such as drones.

The remainder of this article is organized as follows. Section 2 provides information about the telemetry of drones, MEMS gyroscopes, and their hardware imperfections. Section 3 describes our attack model including assumptions. In Section 4, we explain the design of the proposed fingerprinting method. Section 5 provides details of our experiments. Discussions including limitations of the proposed method are presented in Section 6. Section 7 outlines the existing researches on security issues related to our study. We conclude this article with a summary in Section 8.

[1] All previous studies considered fingerprinting with accelerometer output or a combination of accelerometer and gyroscope outputs.

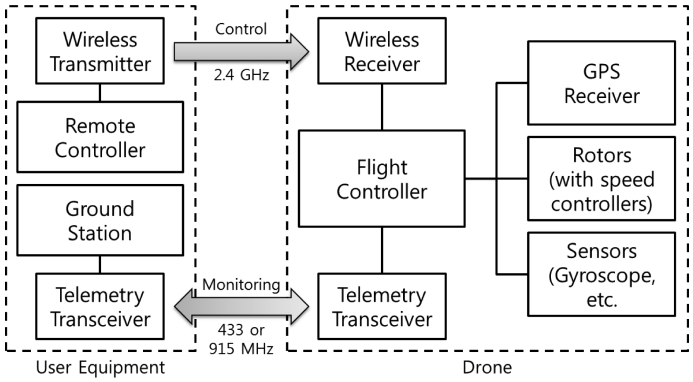


Fig. 1. Block diagram of a typical drone system.

2 BACKGROUND

In this section, we provide information about the telemetry functionality used to monitor drones. We also explain the principle and characteristics of MEMS gyroscopes including the manufacturing mismatch in their hardware that causes the minute difference in the outputs of the MEMS gyroscopes.

2.1 Telemetry in Drones

In a drone, two different types of wireless communication are utilized between the drone and its user. A block diagram of a typical drone is shown in Figure 1. In the middle of this figure, the flight controller, which is the brain of the drone, is connected to a GPS receiver to receive GPS signals from GPS satellites, rotors to adjust their speeds, and sensors[2] to measure the environment. For the wireless communication, the flight controller is also connected to the wireless receiver and the telemetry transceiver. To control the drone, the remote controller wirelessly transmits control signals to the wireless receiver according to user controls. To monitor the status of the drone, both the telemetry transceivers at user and drone sides communicate with each other. The ground station connected a user-side telemetry transceiver comprised of a laptop and monitoring software such as *QGroundControl* [39]. Many telemetry transceiver pairs communicate using frequencies of 433 or 915MHz, and some of them support Wi-Fi.

The most commonly used protocol for the telemetry of drones is the MAVLink protocol [28] defined and built on top of various wireless communication protocols. In MAVLink, the ground station has to send request messages for receiving any data from the drone. According to the predefined MAVLink messages, the ground station can obtain 269 types of data including heartbeat, system status and parameters, sensor outputs, location, and altitude, but the number of supported data types depends on the implementation of each drone. Although the protocol does not support authentication and encryption [11, 23, 27, 41, 43], it has been applied to flight controllers such as *Pixhawk1* [37] and *Pixhawk2* [38], which are used not only for commercial drones, but also for DIY drones. Therefore, if an attacker has the capability to communicate with the telemetry transceiver on a drone using an appropriate radio transceiver, she can collect its status information. In addition, because the range of the wireless communication used in the telemetry transceivers is hundreds of meters, the attacker can collect the status information from the target drones remotely.

[2]These sensors are usually installed on the flight controller board.

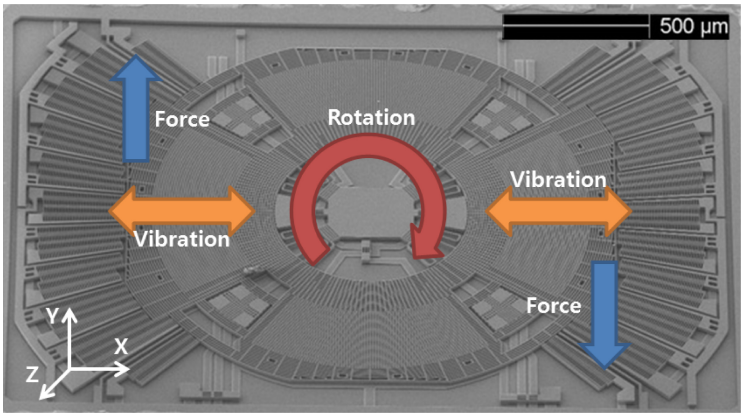


Fig. 2. An example of the structure of a MEMS gyroscope.[3] When a rotation occurs while the middle of the structure is continuously vibrating, a force is generated in a direction perpendicular to the vibrating direction.

2.2 MEMS Gyroscopes

A gyroscope is a device that measures the angular rate when spinning around one or more axes. Based on the angular rate, a system equipped with a gyroscope can calculate the amount of rotation or tilt. Because of the benefit of the MEMS technology that can implement micrometer- or nanometer-sized mechanical structures into a microchip, it is becoming more popular and less expensive. MEMS gyroscopes have been used in many applications, such as smartphones, health care or fitness devices, drones, and gaming mice. Several types of MEMS gyroscopes are available in the market. We differentiate between the analog and digital types according to the representation of the output data and between one-, two-, or three-axis types according to the number of supporting axes. The most popular type is the digital three-axis MEMS gyroscope, which is mostly manufactured by *STMicroelectronics* and *InvenSense*.

A drone is equipped with a MEMS gyroscope that acts as an essential part of their inertial navigation system. To stabilize the drone's movement in the air, its firmware or operating system reads the raw outputs of its sensors, including the MEMS gyroscope, calibrates the outputs, and converts them into meaningful physical quantities to observe and control the posture of the drone.

Operation: The principle of MEMS gyroscopes is a physical phenomenon called the Coriolis effect, which explains the deflected force (called Coriolis force) of a moving object in a rotating structure. An example of the mechanical structure is shown in Figure 2. For sensing a rotation along the Z axis, the structure continuously vibrates along the X direction. When a rotation occurs, the deflected force is generated in the Y direction. This deflected force is measured in terms of capacitances that are sensitively changed by the thickness of the physical gap between two plates in the MEMS structure. By measuring this force, a MEMS gyroscope can sense the angular rate because the force is proportional to it.

Manufacturing Mismatch: Every analog system has variances because of different noise sources in nature, and the mass production of MEMS structures is no exception. Variances exist even among the same models. Since the manufacturing of MEMS sensors is subject to nanometer scales, any small variance leads to manufacturing mismatches [16, 22]. Not all causes of manufacturing mismatches are known, but an obvious cause is related to the capacitance [29]. The electrical capacitance mentioned in the previous subsection is determined by the size of two metallic plates and

[3]The base image is obtained from iFixit website [18].

the gap between them. Therefore, any physical mismatch in the structure of MEMS gyroscopes generates differences in their outputs. These manufacturing mismatches can appear in different features, such as the offset, bias, and skewness of outputs.

2.3 Offsets of MEMS Gyroscopes

A MEMS gyroscope has many electronic and mechanical characteristics. An important characteristic relevant for our fingerprinting method is the *zero-rate level* (also known as the *zero-rate output*) [19–21, 47, 48]. The *zero-rate level* denotes an output level with no actual angular rate. This level of each MEMS gyroscope can vary because of the mechanical stress caused by processes such as soldering and mounting as well as the manufacturing mismatches. It is also known that the *zero-rate level* is not affected greatly by the changes in temperature and does not change much over time. We call this characteristic as *offset* in this article.

In many systems that utilize MEMS gyroscopes, the raw sensor outputs are calibrated or compensated for standardized measurements. Because the measurement of MEMS gyroscopes is affected by the temperature and the manufacturing mismatches, the systems compensate for the variance in temperature and calibrate the output to be zero with no rotations. Through this calibration process, the sensitivity and the offset of MEMS gyroscope outputs are adjusted properly so that the calibrated outputs of every MEMS gyroscope have similar levels for the same physical quantities. Although some variances are removed by compensation and calibration, previous studies [9, 10] proved that the outputs of each sensor still have distinguishable variances.

As an example of the calibration, according to open-source drone projects [13, 14] that support various MEMS gyroscopes, the relationship between the raw outputs $d_{raw}[n]$ and the calibrated outputs $d_{cal}[n]$ of a MEMS gyroscope for each axis (an index for axes is omitted) is modeled as follows:

$$d_{cal}[n] = A \times d_{raw}[n] - B, \tag{1}$$

where A is a sensitivity factor for converting the measured data into an actual physical quantity, B is a bias factor used to make the offset of the calibrated outputs close to zero, and n is a discrete time index for collected N samples ($0 \leq i \leq N-1$). A is a constant value that is unique for each model of MEMS gyroscopes [13, 14]. B is unique for each MEMS gyroscope, and its value is determined by the calibration process during the booting stage of the device in a stationary state [13, 14]. When the average value of the raw outputs is $d_{raw}|_{avg} = \frac{\sum_{n=0}^{N-1} d_{raw}[n]}{N}$, the raw outputs can be represented as the sum of the alternating $d_{raw}[n]|_{alt}$ (i.e., $d_{raw}[n] - d_{raw}|_{avg}$) and the average $d_{raw}|_{avg}$ components. Then, from Equation (1), we obtain the following:

$$d_{raw}[n] = d_{raw}[n]|_{alt} + d_{raw}|_{avg} = \frac{d_{cal}[n]}{A} + \frac{B}{A}. \tag{2}$$

When the device is calibrated in a stationary state, because both $d_{raw}[n]|_{alt}$ and $d_{cal}[n]$ are close to zero, the average component $d_{raw}|_{avg}$ is equal to $\frac{B}{A}$, which is the *offset* of the raw outputs. Therefore, by averaging the raw outputs, the offset can be easily calculated.

2.4 Obtaining the Outputs of MEMS Gyroscopes

In the case of drones, the outputs of the MEMS gyroscopes, which are calibrated by the flight controller, can be obtained using the telemetry. Even though each firmware developer may adopt different implementations, the calibrated outputs are usually included in the status information, because they can be used for not only monitoring but also debugging. For most flight controllers, the offsets of MEMS gyroscopes obtained during the boot time are provided as well.

On the other hand, in the case of stand-alone MEMS gyroscopes, we can access the MEMS gyroscopes directly by using a programmable processor such as Arduino, which allows the collection of

raw outputs. Analog MEMS gyroscopes represent the angular rate by means of the output voltage level, and thus the processor only needs to read this level. Most of the digital MEMS gyroscopes support a digital communication interface called by the serial peripheral interface (SPI). By sending the request messages through the SPI, the processor can receive response messages including the values of the angular rate as a bit sequence.

3 ATTACK MODEL

In this section, we introduce our attack model, including the assumptions for our fingerprinting method for tracking drones. Based on our attack model, we also suggest a possible attack scenario.

3.1 Target System

In this article, we focus on fingerprinting and tracking flying drones based on the offsets from the MEMS gyroscopes and location information included in the status information that can be obtained through open telemetries. For this, we need to investigate if there exists a useful and persistent identifier used in telemetry and its underlying communication. As discussed in Section 2.1, status information does not include any identifier. Moreover, the network address of the telemetry transceivers may not be used for reliable fingerprints because they can be temporarily assigned and periodically updated, and even the telemetry transceivers can be easily replaced because they are usually stand-alone modules just connected to the mainboard of drones unlike the MEMS gyroscopes which are soldered on the mainboard. Hence, the manufacturing mismatches of the MEMS gyroscopes are one of the few characteristics that can serve as a fingerprint for drones.

We make the following assumptions regarding the target systems:

- —The target systems are drones equipped with MEMS gyroscopes and telemetry transceivers that have no authentication and encryption.
- —The target systems are implemented to transmit the location information (or GPS data) and the offsets (or the raw and calibrated outputs) of MEMS gyroscopes through telemetry.

A MEMS gyroscope is the essential component of the IMU that controls a drone's attitude. Telemetry transceivers are optional components, but they are necessary not only for monitoring but also for configuring some parameters wirelessly. The lack of authentication and encryption of such transceivers has been publicly reported [11, 23, 27, 41, 43]. Some drones use Wi-Fi modules to communicate with user controllers and with ground stations as a telemetry transceiver as well (e.g., 3DR Solo [1]). Wi-Fi supports authentication, but it also has vulnerabilities such as the use of default passphrases. We also confirmed that the location information (i.e., latitude and longitude) and the scaled offsets of MEMS gyroscopes can be obtained in default from two kinds of popular flight controllers, Pixhawk1 [37] and Pixhawk2 [38]. Therefore, the first and second assumptions hold in practice.

3.2 Capabilities of Attacker

To identify and track target drones, we assume that an attacker (1) has one or multiple ground stations using the same type of telemetry transceivers to the target systems or software-defined radio devices, such as the universal-software radio peripheral, HackRF, and BladeRF, using which, the attacker implements the telemetry transceivers and (2) can obtain the location information and offsets of MEMS gyroscopes from the target drones.

The offsets can be obtained in three ways. First, if the offsets are readily available through the telemetry, the attacker only needs to send requests to obtain them. Second, if the attacker can acquire the raw outputs of MEMS gyroscopes when the target drones are not moving (i.e., before flying), the offsets can be simply calculated by averaging the raw outputs (as explained in

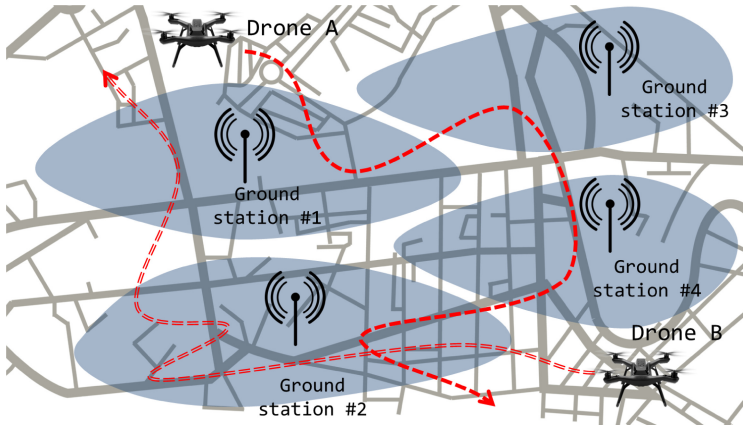


Fig. 3. An attack scenario for tracking drones using multiple ground stations based on the proposed fingerprinting and tracking method. (Each shaded region denotes the coverage of a station.)

Section 2.3). However, this requires multiple samples of the raw outputs collected over several seconds. Third, if both the raw and calibrated outputs of MEMS gyroscopes can be acquired and synchronized, the offsets can be calculated by comparing them, regardless of whether the drones are flying, because both are highly correlated [Equation (1)].

3.3 Attack Scenario

Under the assumptions mentioned above, an attacker can collect the status information via telemetry interfaces from multiple drones using a single ground station because the range of the wireless communication using the telemetry is hundreds of meters. From the collected information, the attacker can know the locations of the drones, but the location data cannot be assigned to specific drones, owing to the lack of identifiers in the collected information. However, the offsets of MEMS gyroscopes in the information can be used as fingerprints, allowing the attacker to match the locations and the drones.

The range of wireless communications can be increased to several kilometers by using powerful COTS radio frequency amplifiers [49]. Therefore, the attacker can track the target drones in large areas, and the coverage of the tracking can be widened by using multiple ground stations like mobile cell towers. Figure 3 illustrates this attack scenario.

4 FINGERPRINT DESIGN

In this section, we describe our investigation to find the fingerprints, including the evidence of the fingerprints in the measured data from several drones and the detailed design of our fingerprinting method. Through our measurements and analyses, we conclude that the offsets of MEMS gyroscopes for each axis can serve as efficient fingerprints.

4.1 Evidence of Fingerprints

To ensure that offsets can be used for fingerprinting drones, we first evaluate with a number of commercial drones equipped with commercial flight controllers, both of whose source codes are open. As examples, we collected the offset data included in the status information from five drones (four Pixhawk1 [37] and one Pixhawk2 [38] flight controllers based on the ArduPilot project) with the latest firmware. These drones transmit scaled offsets (i.e., B as explained in Section 2.3) of

Table 2. Offsets Collected from Five Drones at Room Temperature

Drone (flight controller)	Offset of X axis	Offset of Y axis	Offset of Z axis
Pixhawk1 #1	−12.57	65.75	4.07
Pixhawk1 #2	8.32	34.46	−1.15
Pixhawk1 #3	−0.81	52.28	−9.53
Pixhawk1 #4	22.95	19.57	−15.75
Pixhawk2	−4.26	−12.06	3.70

MEMS gyroscopes in the status information using the telemetry transceiver [13, 14]. In this case, we can possibly use these offsets directly to fingerprint them even though they are scaled. However, since sensitivity factors may be different from each gyroscope model, to accurately compute the actual offset, we need to find the sensitivity factor for each gyroscope, which can be obtained from the source code of flight controller. For the MPU6000 gyroscope with which the five drones are equipped, the constant sensitivity factor, A, is $\frac{0.0174532}{16.4}$ [13, 14]. The actual offsets (i.e., $\frac{B}{A}$) for all axes are listed in Table 2 (the offsets are averaged for 10 times of measurements at room temperature), and we can see that they are distinguishable. These values are determined during the boot procedure before the drone's flight and are not changed during the flight.

4.2 Large-Scale Experiments

For further experiments and evaluations, we chose tens of stand-alone MEMS gyroscopes to test instead of commercial drones because our method is based on the outputs of MEMS gyroscopes to identify the drones, and purchasing tens of commercial drones is not economical. To measure offsets from stand-alone MEMS gyroscopes, we need to collect their raw outputs as explained in Section 2.3. The remaining part of this section explains the experiments, and Section 5 reports evaluation results performed using those stand-alone MEMS gyroscopes.

An overview of our fingerprinting method for stand-alone MEMS gyroscopes is illustrated in Figure 4. The upper and the lower parts are for the learning and the matching processes. Both processes consist of three steps: (1) collecting the raw data, (2) generating fingerprints, and (3) storing the fingerprints in the attacker's database (in the learning process) and searching the fingerprints of collected samples among the existing fingerprints (in the matching process).

4.2.1 Collecting Raw Data. All of the MEMS gyroscopes that we investigated are digital. To read the raw outputs, we built an SPI-to-USB converter using Arduino. This converter reads the raw outputs for the three axes every 10 milliseconds and transmits the outputs to a laptop. A program on the laptop receives the outputs and stores them into files. The setting for collecting the raw data is shown in Figure 5.

We investigated 70 MEMS gyroscopes listed in Table 3 and collected 2^{19} samples (approximately 524k samples) for all axes of each MEMS gyroscope when they were on a stationary desk. After a month, we collected additional 2^{16} samples (approximately 65k samples) under identical conditions to test whether the fingerprints are time-invariant. The second collection was obtained after shaking the MEMS gyroscopes randomly for 30 minutes, imitating their normal usages. This movement is automatically generated using a DIY gimbal shown in Figure 6, which can perform random rotations around the three axes independently using three identical Arduino boards and step motors. All of our measurements are performed in the same environment (in a closed room at room temperature).

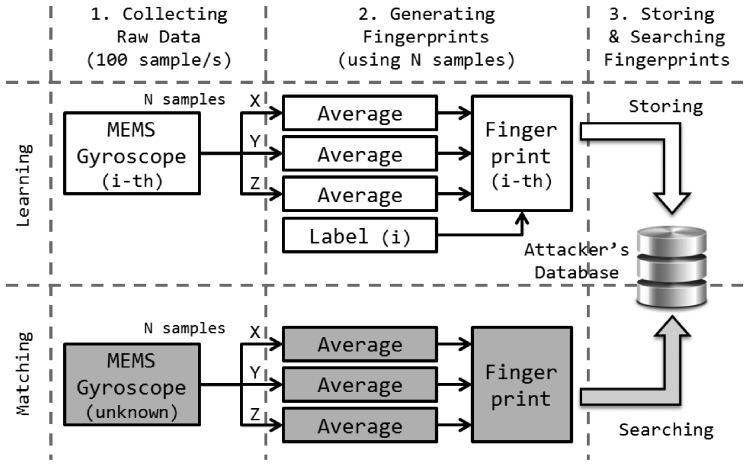


Fig. 4. An overview of the proposed fingerprinting method (for large-scale evaluation using stand-alone MEMS gyroscopes).

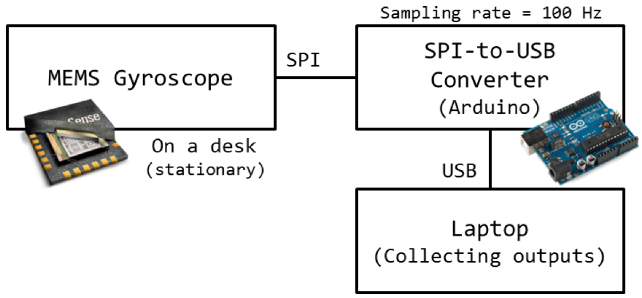


Fig. 5. Setting to collect the raw outputs of MEMS gyroscopes (5 different models and 70 chips).

Table 3. List of MEMS Gyroscopes Used in Our Experiments

Model name	# of chips	Manufacturer
L3G4200D	11	STMicroelectronics
L3GD20	12	STMicroelectronics
MPU6050	17	InvenSense
MPU6500	16	InvenSense
MPU9150	14	InvenSense

4.2.2 Generating Fingerprints. To illustrate that the offsets can be used as fingerprints, the offsets of the three axes for all collected samples are shown in Figure 7. Each point is the average of all samples per axis, and thus 70 points are represented (i.e., one point per MEMS gyroscope). We can observe that the points are distinguishable and rarely overlapped. Based on this observation, we calculate the offsets as a fingerprint of the i-th MEMS gyroscope F_i, where $0 \leq i \leq 69$, by averaging the raw outputs for the three axes $d_{i,x}[n]$, $d_{i,y}[n]$, and $d_{i,z}[n]$:

$$F_i = \left[\sum_n \frac{d_{i,x}[n]}{N} \quad \sum_n \frac{d_{i,y}[n]}{N} \quad \sum_n \frac{d_{i,z}[n]}{N} \right]. \tag{3}$$

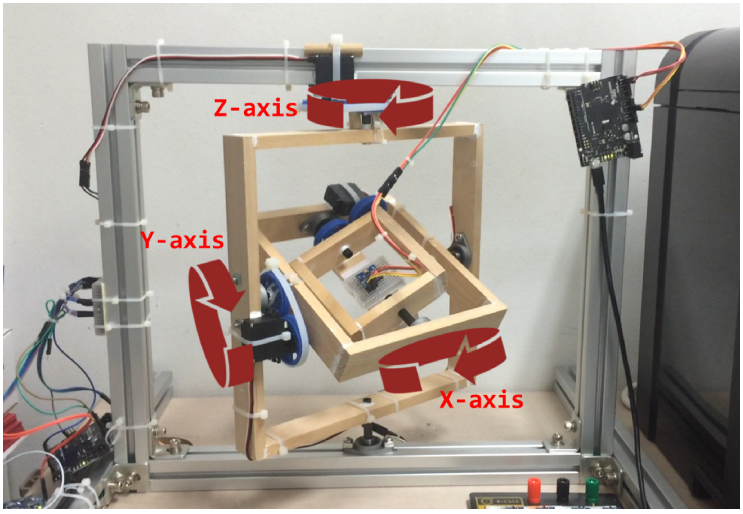


Fig. 6. A DIY gimbal that can make random rotations in three axes independently using three Arduino boards and step motors.

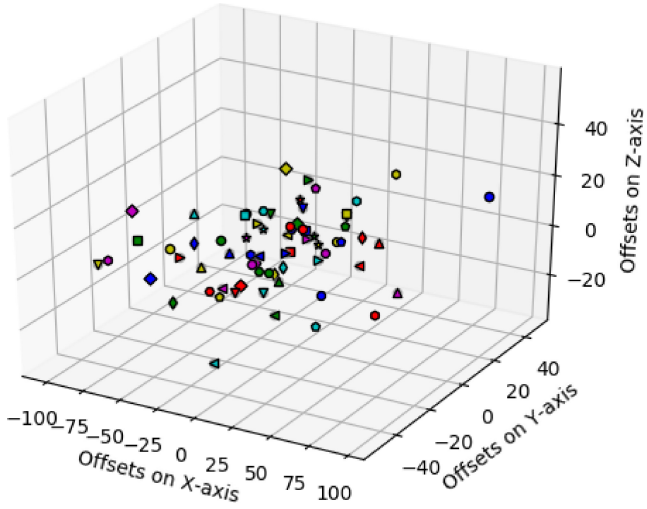


Fig. 7. Spatial distribution of the offsets for three axes. (Each point is averaged for all samples per axis and represents each MEMS gyroscope.)

In this equation, n is a set of discrete time indices that includes samples in a specific or the entire time interval, and N is the number of corresponding samples. Finally, a fingerprint of one MEMS gyroscope is represented as a point in a three-dimensional coordinate system.

Because our fingerprints are mapped into points in space for all MEMS gyroscopes, we can calculate the Euclidean distances between fingerprints. Figure 8 shows the cumulative distribution of the Euclidean distances between two fingerprints. In our measurements, about 80% of the differences in distance are over 30, and this fact strengthens the argument that the offsets can be used as effective fingerprints. The minimum and maximum differences are 2.94 and 207.46, respectively. (Figures 7 and 8 show the distribution of the intra-distance among the fingerprints from all 70

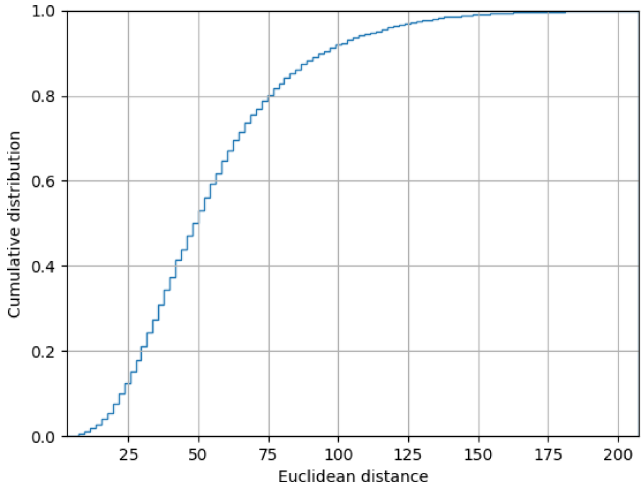


Fig. 8. A cumulative distribution of the Euclidean distances between two fingerprints. (The minimum Euclidean distance is 2.94, and the maximum is 207.46.)

MEMS gyroscopes. For more information, the distributions of the inter-distance for each gyroscope models are described in Appendix A.)

4.2.3 Storing and Searching Fingerprints. The fingerprints generated by Equation (3) are assumed to be stored and labeled in an attacker's database. When a new fingerprint F_p is obtained, the attacker can identify a MEMS gyroscope (i.e., a drone) by taking the i-th fingerprint F_i that has the minimum Euclidean distance from a given offsets:

$$\arg\min_i \|F_p - F_i\|. \tag{4}$$

However, using only the minimum Euclidean distance for matching makes it impossible to find a new distinct device. For example, the matching might identify a point that lies outside the space shown in Figure 7 as a fingerprint from within the space. To avoid this, we need to determine a decision boundary for each fingerprint. We assume virtual spherical and non-overlapping boundaries around each fingerprint. Accordingly, we chose the half of the Euclidean distance between i-th fingerprint and its nearest neighboring fingerprint for the bound B_i of the i-th fingerprint as shown in Equation (5). Now, Equation (4) can be modified as Equation (6). Using this equation, we can detect new devices whose fingerprints lie beyond the spherical boundaries:

$$B_i = \min_j \frac{\|F_i - F_j\|}{2}, \tag{5}$$

$$\arg_i (\|F_p - F_i\| < B_i). \tag{6}$$

The bound B_i can be adjusted according to the density of the fingerprints. When an attacker starts generating fingerprints, she starts with one or only a small number of fingerprints. In this situation (i.e., when the density is low), the bound B_i, which is determined by Equation (5), is not suitable because it is too large. Therefore, we need to reduce the bound appropriately until a sufficient number of fingerprints are generated. For instance, the bound can be fixed in this case as a constant considering the minimum difference in the distance among fingerprints (2.94 in our dataset). The bound can also be dynamically determined to cluster the collected offsets to minimize the total standard deviation of the offsets on the basis of each fingerprint.

5 PERFORMANCE EVALUATION

We evaluated the proposed fingerprinting method on a large scale by using several metrics. Additionally, we investigated the invariance of our fingerprints with temperature and time.

5.1 Experimental Setup

The experiment consists of two steps, namely training and matching. We divided the 2^{19} samples that we collected for each axis into $\frac{2^{19}}{N}$ groups (i.e., one group consists of continuous N samples for each axis) and obtained the offsets for each group.

For training, we randomly selected one group and generated fingerprints by averaging the samples in that group. To determine the minimum number of samples required for efficiently generating fingerprints, we changed the number of samples N in a group from 2 to 1,024, exponentially. As inputs for the matching step, we also randomly selected 500 groups among the remaining 511 groups (i.e., except the one group used for training).[4] The fingerprints of the inputs in the matching step are generated in the same manner to those in the training step, and B_i is calculated by Equation (5). In both steps, we tested the performance of the fingerprints using the offsets of three axes, two axes, and one axis. Each test was repeated 100 times, and the results are averaged.

5.2 Evaluation Metrics

In the previous device fingerprint studies, several evaluation metrics such as *precision*, *recall*, *F-score*, and *accuracy* were used. According to the fingerprint result for the i-th device, each fingerprint event can be divided into four cases, namely, the true positive TP_i, the true negative TN_i, the false positive FP_i, and the false negative FN_i. Then, the precision PR_i, recall RE_i, F-score FS_i, and accuracy ACC_i of the i-th device are defined as follows:

$$PR_i = TP_i/(TP_i + FP_i), \tag{7}$$

$$RE_i = TP_i/(TP_i + FN_i), \tag{8}$$

$$FS_i = (2 * PR_i * RE_i)/(PR_i + RE_i), \tag{9}$$

$$ACC_i = (TP_i + TN_i)/(TP_i + FP_i + FN_i + TN_i). \tag{10}$$

Precision, which is also known as *positive predictive value*, indicates how many fingerprinted devices are relevant. *Recall* is also called *sensitivity* or *hit rate*, and it signifies how many relevant devices are fingerprinted. *F-score* is the harmonic mean of precision and recall, and *accuracy* is the proportion of correct fingerprints. In a multi-class classification, considering the i-th device, all the results that are estimated to be other devices for other devices are true negative, even though they are not correct. This indicates that *accuracy* is not a good measure for this experiment; therefore, we chose *precision*, *recall*, and *F-score* to evaluate the proposed fingerprinting method.

The overall performance of the proposed fingerprinting method is represented as the averaged precision PR, recall RE, and F-score FS for all classes. They are defined as the following equations. The number of all devices M is 70 in the evaluation.

$$PR = \sum_{i=0}^{M-1} \frac{PR_i[i]}{M}, \tag{11}$$

$$RE = \sum_{i=0}^{M-1} \frac{RE_i[i]}{M}, \tag{12}$$

$$FS = (2 * PR * RE)/(PR + RE). \tag{13}$$

[4]When N is 1,024, the total number of groups is 512.

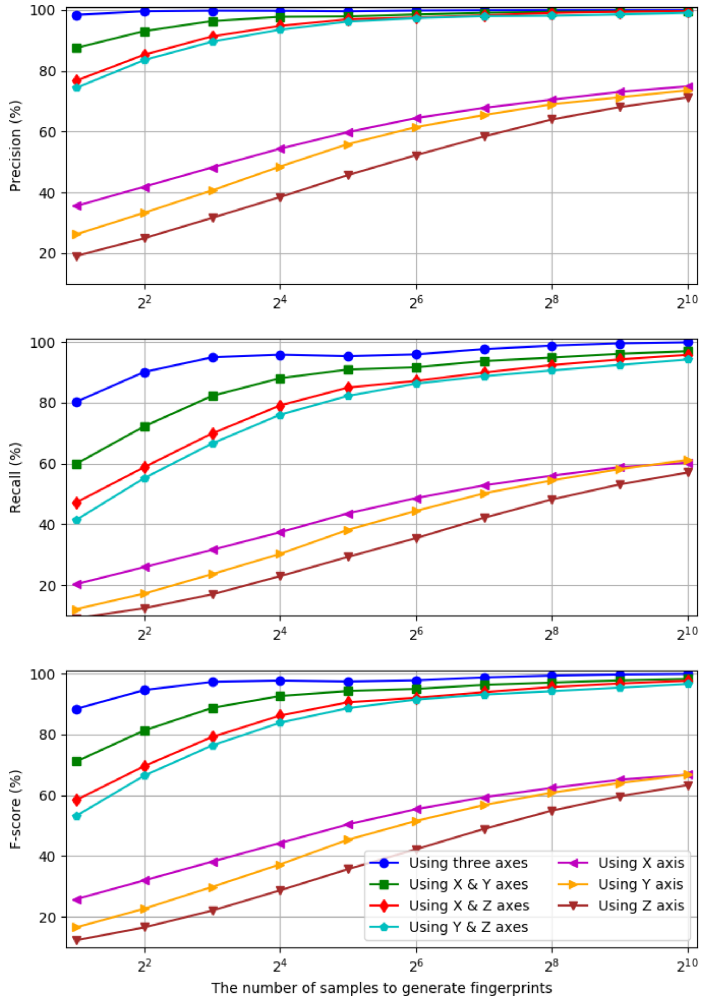


Fig. 9. The precision, recall, and F-score of the proposed fingerprinting method. (When the offsets of all the axes with 128 samples are used as fingerprints, the F-score is close to 98.78%. Under the same condition, the F-score is close to 94.47% when using the offsets of only two axes.)

5.3 Overall Performance

To identify which MEMS gyroscope the data is sourced from, we need to find a fingerprint that satisfies Equation (6). The performance metrics defined in Section 5.2 are measured and shown in Figure 9 as the results. In all cases, as N increased, all metrics tend to be increased and converge (the X axis of Figure 9 is in the log scale). The values of the metrics when they converge (i.e., for $N = 1024$) are summarized in Table 4. In our experiment, the F-scores that we achieved are 99.94% and 97.52% when using the offsets of three and two axes, respectively.

The results in Figure 9 show that the larger N we use for generating fingerprints, the better performance we can achieve. The larger N requires more time to fingerprint because an attacker has to collect N samples while the target devices are stationary. Therefore, the efficiency of the fingerprinting and N present a trade-off, and we need to minimize N while maintaining the performance sufficiently. When we chose N as 128, the F-scores reach 98.78% and 94.47% when using the offsets of three and two axes, respectively. The detailed results are listed in Table 5.

Table 4. Results of the Performance Evaluation When Using 1,024 Samples to Generate Fingerprints (the Convergence Values of Each Metric)

Condition	Precision	Recall	F-score
Using three axes	100.0%	99.88%	99.94%
Using two axes (in average)	99.48%	95.64%	97.52%
Using one axis (in average)	73.26%	59.47%	65.65%

Table 5. Results of the Performance Evaluation When Using 128 Samples to Generate Fingerprints

Condition	Precision	Recall	F-score
Using three axes	99.95%	97.63%	98.78%
Using two axes (in average)	98.47%	90.79%	94.47%
Using one axis (in average)	63.90%	48.35%	55.03%

Based on the results, we can conclude that if an attacker obtains the offsets of MEMS gyroscopes from drones, she can efficiently identify the drones. Furthermore, in our setting, if an attacker can read simultaneously the raw and calibrated outputs of a MEMS gyroscope, she needs only approximately 1.28 seconds (128 samples × 10 milliseconds per sample) to generate the fingerprint of a drone and to identify it.

5.4 Time Invariance

The datasheets [19–21, 47, 48] specify that the MEMS gyroscopes are tolerant of temperature and time variations. While the variances over temperature are described in the datasheets, the variances over time are not listed.

To examine the time invariance of the fingerprints, we collected the second dataset after one month as explained in Section 4.2.1. We measure the variance v_i of the i-th MEMS gyroscope as the ratio of the Euclidean distance between the original fingerprint F_i and the changed offsets F_i' over time to the Euclidean distance between the original fingerprint and the one nearest to it. If the variance exceeds 1.0, our method determines that the changed offsets originate from a different MEMS gyroscope, which means that our fingerprint is time-variant. The variance of the i-th MEMS gyroscope is represented by the following equation:

$$v_i = \frac{\left\| F_i - F_i' \right\|}{B_i}. \tag{14}$$

The results of our investigation are shown in Figure 10. All variances are below 0.5, which means that the generated fingerprints are still valid after one month.

Figure 11 shows the precision, recall, and F-score of the data after one month. For the experiment, we used the same setup and fingerprints, but tested 60 groups randomly selected from the secondary dataset because the minimum number of groups is 63 when N is 1,024. This test was also repeated 100 times. For all cases, the evaluation metrics show a tendency similar to that of the results in Section 5.3, but their values are decreased over time. When we use the offsets of all axes and N is 1,024, we achieved an F-score of 97.91%. The detailed results are summarized in Table 6.

The evaluation metrics corresponding to 128 samples for generating fingerprints are compiled in Table 7. When we use the offsets of all axes, the F-score is 96.58%, which is still effective, but dropped to 78.45% when using the offsets of two axes.

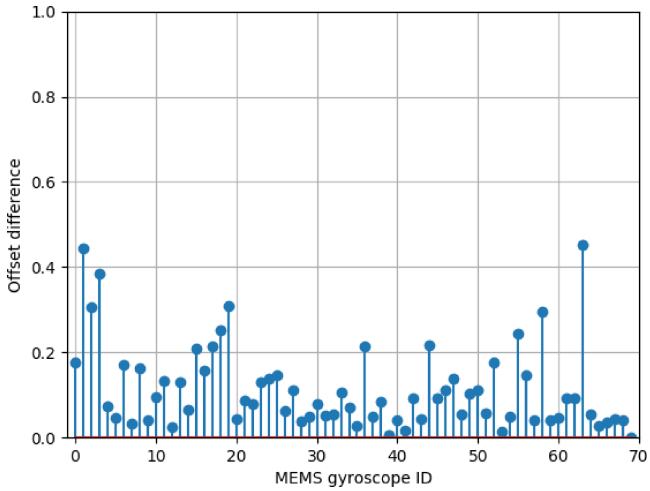


Fig. 10. The amount of change in offsets after a month: the ratio of the changed Euclidean distance to the half of the minimum Euclidean distance between the one fingerprint and others. (The IDs are assigned sequentially in the order shown in Table 3. If the ratio exceeds 1.0, the changed offsets will be judged as different devices.)

5.5 The Effect of Temperature

The MEMS gyroscopes can be impacted by thermal noise because they are fabricated with silicon, which is a temperature-sensitive material.

To investigate the effect of temperature, unlike the previous experiments evaluating 70 MEMS gyroscope sensors, we used five flight controllers (i.e., drones). We simulated real-world environments with varying ambient air temperature against these five drones. On average, the room temperature was 24.0°C. To control the temperature, we put the flight controllers in a refrigerator and in a sunny spot for an hour. We measured the offsets from the devices while we repeatedly reduced and raised the temperature 10 times. In the measurements, the average ambient air temperatures were 4.0 and 29.6°C, respectively.

The average and the standard deviation of the amount of changes in Euclidean distances between offsets are listed in Table 8. When the ambient air temperature is decreased from 24.0 to 4.0°C and increased from 24.0 to 29.6°C (the second and third columns in Table 8), the average changes of measured fingerprints in Euclidean distance are relatively small considering the cumulative distribution shown in Figure 8. Additionally, when the temperature went back to 24.0°C (i.e., room temperature) after variations (the fourth column in Table 8), we can see that the fingerprints also went to near the original points as shown in Figure 12 with a relatively small amount of changes, comparing to the changes in decreasing or increasing temperature cases. These results mean that the fingerprints are not much affected by temperature variations, but they can probabilistically lead misclassification for fingerprinting.

To examine how these results affect the classification, we assume an attacker who has 75 fingerprints: 70 from Figure 7 and 5 by averaging points in Figure 12 (at 24.0°C). Then, we tested whether each measured offset (total of 50 measurements, as we measured 10 times for 5 drones) changed by temperature variance can be correctly matched to one of those 75 fingerprints. As results, after the temperature raised to 29.6°C, 45 measurements of the changed fingerprints are matched correctly (i.e., 90%). When the temperature dropped to 4.0°C, the number of the correct match is 39 (i.e., 78%). In detail, especially, Pixhawk2's success rate is very low for both 29.6 and

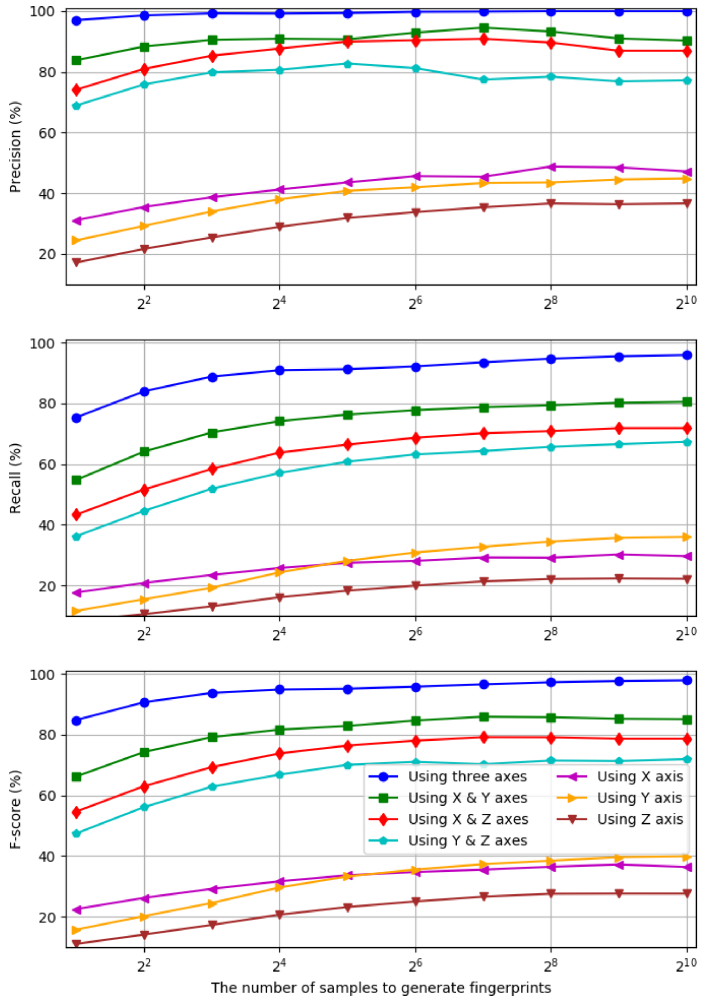


Fig. 11. The precision, recall, and F-score of the proposed fingerprinting method after one month. (When the offsets of all axes with 128 samples are used as fingerprints, the F-score is close to 96.58%. In all cases, the F-score is decreased over time.)

4.0°C cases (each had 50 and 10% success rate) because its fingerprints are located in a dense area. In all cases of Pixhawk2, they were ranked top 5 among closest offsets in terms of the distance. Hence, even though the offsets are affected by temperature, they are still useful as fingerprints to identify the drones within a specific range of temperature.

Considering the results in Section 5.4 and 5.5, the proposed method can be used as a persistent, robust, and effective fingerprinting method even when considering the variances of temperature and time. However, the results also indicate that an attacker may need to update the fingerprints over these variations because the changes can be biased in a specific direction cumulatively.

6 DISCUSSION

In this section, we discuss the possibility to extend the proposed fingerprinting method to drones from other manufacturers and for drone traffic management systems such as UTM. We also discuss the limitations and a possible countermeasure of the proposed fingerprinting method.

Table 6. Results of the Performance Evaluation after a Month When Using 1,024 Samples to Generate Fingerprints (the Convergence Values of Each Metric)

Condition	Precision	Recall	F-score
Using three axes	100.0%	95.90%	97.91%
Using two axes (in average)	84.83%	73.19%	78.56%
Using one axis (in average)	42.91%	29.25%	34.65%

Table 7. Results of the Performance Evaluation after a Month When Using 128 Samples to Generate Fingerprints

Condition	Precision	Recall	F-score
Using three axes	99.87%	93.49%	96.58%
Using two axes (in average)	87.66%	71.03%	78.45%
Using one axis (in average)	41.41%	27.74%	33.15%

Table 8. The Average of Changes in Euclidean Distance between Fingerprints According to Temperature Variations (the Numbers in Round Brackets Are the Standard Deviations of the Changes)

Drone (flight controller)	24.0 ⇒ 4.0°C	24.0 ⇒ 29.6°C	24.0 ⇒↑↓⇒ 24.0°C
Pixhawk1 #1	15.48 (3.98)	7.23 (4.35)	4.53 (3.73)
Pixhawk1 #2	6.36 (1.80)	4.44 (2.52)	1.65 (0.66)
Pixhawk1 #3	6.95 (1.14)	6.10 (3.59)	2.45 (1.41)
Pixhawk1 #4	3.37 (0.61)	2.71 (1.17)	1.36 (0.54)
Pixhawk2	14.50 (2.76)	6.75 (4.05)	1.79 (0.99)

6.1 Extensions

We show that, using the proposed method, fingerprinting drones and tracking their locations are possible for two kinds of commercial flight controllers, i.e., Pixhawk1 [37] and Pixhawk2 [38], which are used for the drones manufactured by 3DR[5] as well as for many DIY drones and open-source drone projects. If an attacker can obtain the offsets of MEMS gyroscopes from any other model of drones, the same approach can be applicable. However, because the sensitivity factor (in Section 4.1) may be different per model, the density of fingerprints in three-dimensional space may also differ per model. Thus, the fingerprinting performance can decrease unless the exact value of the sensitivity factor is known. We note that the drones manufactured by DJI[5] use a proprietary telemetry protocol, and its state information does not include the offsets per axis.

The proposed fingerprinting and tracking method can also be used for drone traffic management systems such as UTM mentioned in Section 1. In UTM, all drones should be registered, and their users should submit their detailed flight plans in advance. Then, the system checks possible conflicts, approves (or disapproves) the plans, and tracks the drones. Tracking the registered drones in UTM is based on low-altitude radar, cellular signals, ADS-B, and GPS information by the UAS ground control station [53]. In the registration and submission steps, the offsets of MEMS gyroscopes from each drone can be measured legitimately and used as an identity of the drone.

[5]DJI and 3DR are drone manufacturers that have the first and third market shares, respectively [55].

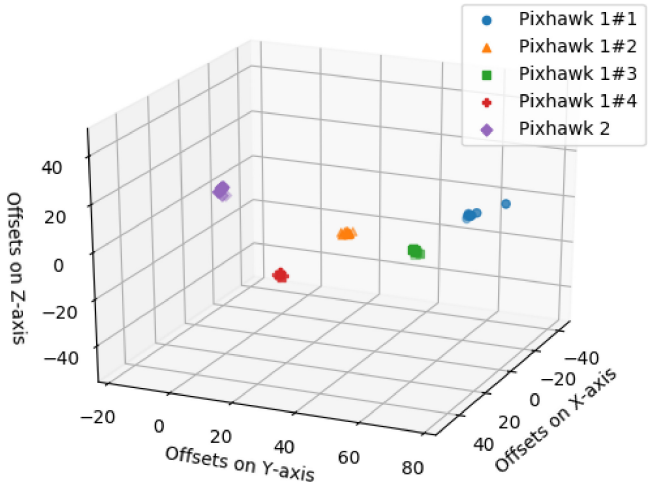


Fig. 12. Spatial distribution of the offsets when the temperature went back to room temperature after variations.

Therefore, the proposed method can be used to track moving drones without any modification for them and with no additional cost for drone management systems.

6.2 Limitations

Because the probability distribution of the offsets from every MEMS gyroscope is unknown, we can only calculate the entropy of the offsets from our sample datasets. More specifically, based on our dataset used in Figures 7 and 8, we can roughly estimate the entropy of our fingerprints with several assumptions. For this estimation, we assume that (1) the outputs of MEMS gyroscopes from each axis are independent to other axes, (2) the offsets are uniformly distributed in the range (r_X, r_Y, and r_Z) between the maximum and minimum values in our dataset for each axis, and (3) the interval of the uniform distribution is 2.94 (the minimum Euclidean distance in Figure 8). These assumptions are inevitable considering the size of our dataset.

When the probability distributions of X, Y, and Z axes are $P_X[i]$, $P_Y[i]$, and $P_Z[i]$ (i is an index of an interval in each axis, and the total number of intervals is n_X, n_Y, and n_Z for each axis.), by the assumptions, n_X is $r_X/2.94$, and $P_X[i]$ is $\frac{1}{n_X}$ (the same for other axes). Then, the entropy of the fingerprints can be obtained as follows:

$$H = -\sum_{i=1}^{n_X} P_X[i] \log P_X[i] - \sum_{i=1}^{n_Y} P_Y[i] \log P_Y[i] - \sum_{i=1}^{n_Z} P_Z[i] \log P_Z[i]. \tag{15}$$

By Equation (15), the estimated entropy of the fingerprints collected in our experiments is 16.04. Therefore, increasing the number of devices that need to be identified decreases the distances among the fingerprints, while the amount of noise does not change. However, we note that the performance degradation caused by an increasing number of devices applies to other fingerprint techniques as well.

6.3 Countermeasure

The root cause that makes the fingerprinting and tracking possible is the lack of authentication and encryption not only in MAVLink protocol, but also in wireless communication protocols for

the telemetry. However, MAVLink protocol is used as the upper layer of various wireless communication protocols. Therefore, using secure wireless communication protocols is an obvious countermeasure without modifying MAVLink protocol. We note that, today, most of the wireless communication mechanisms for telemetry are not secure.

7 RELATED WORK

Security research on sensor-related issues is a relatively new field that gained popularity with the rise of embedded devices such as smartphone and Internet of Things devices. This study focuses on designing a fingerprinting mechanism based on sensor data for tracking the locations of drones. In this section, we review the existing studies on sensor-based device fingerprints and other privacy problems caused by sensors.

7.1 Device Fingerprinting Using Sensors

There exist quite a few studies focusing on device fingerprinting methods using MEMS sensors. Bojinov et al. [4] attempted to identify mobile devices on a large scale by using the hardware imperfections of MEMS accelerometers. They collected sensor outputs through *web browsers* of smart devices, and their results showed that 33 and 3,583 smart devices in their lab and a public place could be classified with accuracies of 100% and 15.1%, respectively. AccelPrint [10] also used MEMS accelerometers, and achieved an accuracy of 99.9% using supervised machine learning for the outputs of MEMS accelerometers collected through *web browsers* of 27 smart devices and SPI interfaces of 80 accelerometer chips. Most recently, Das et al. [9] conducted experiments similar to those of AccelPrint, but they used both MEMS accelerometers and gyroscopes. They collected data from 30 and 63 smartphones in the lab and a public place, and the F-score results were 99% and 95%, respectively.

In contrast, our work is different from these works in several ways. (1) They used accelerometers or combinations of a gyroscope and accelerometer, while we used only gyroscopes. (2) Our raw output samples can be obtained in 1.28 seconds with comparable accuracy, while previous studies require samples longer than 30 seconds. In addition, when we gather the calculated offsets through telemetry, only a few milliseconds are required for exchanging several telemetry messages. (3) Our method can be applied regardless of whether the target is moving or stationary.

In addition to MEMS sensors, microphones or speaker and microphone pairs (i.e., acoustic sensors) were used to identify smart devices [8, 25, 57]. Analog signals and hardware modules to receive these signals (i.e., wired or wireless analog signal sensors) were also used for device fingerprinting [2, 7, 12, 24].

Furthermore, physical unclonable functions (PUFs) can be utilized for device fingerprinting because of their inter-device uniqueness. In 2010, Rosenfeld et al. proposed the first sensor-based PUF using non-homogeneous coatings on sensor chips that causes uniqueness in measurements [40]. The unclonable physical features of touchscreen sensors and MEMS gyroscopes were also used for PUFs [42, 52]. However, these sensor-based PUFs are implemented in advance on the target devices. Especially, the MEMS-based PUF [52] requires wafer-level hardware modification. Therefore, these techniques are not appropriate for our case.

7.2 Other Privacy Issues Caused by Sensors

The sensors of smart devices such as accelerometers, gyroscopes, microphones, and cameras have been used to obtain private information. Many researchers have been interested in problems related to leakage of keystrokes, passwords, and PIN numbers. Previous studies showed that an

attacker can successfully infer those text data from sensor outputs collected by malicious applications [3, 5, 6, 26, 33, 34, 36, 44, 46, 51, 56] or web browsers [30, 31]. It is also known that sensors can extract the voice of an adjacent user and the locations of devices. Gyrophone [32] used MEMS gyroscopes to eavesdrop on a conversation near a smartphone. Accomplice [15] used MEMS accelerometers to infer the location of smartphones. Barometers have also been used to infer or track locations [17, 35].

8 CONCLUSION

Recently, drones have become increasingly popular for various applications. In terms of security, maliciously tracking the locations of drones is a serious threat to the drone operators because it can cause escape from drone surveillance, service disturbance, delivery information leakage, and drone capture by attackers. Drones transmit their status information through telemetry channel based on the MAVLink protocol upon wireless communication protocols, and the information contains the location, the offsets of MEMS gyroscopes, and other sensor-related data. Because most of the telemetry communication supports neither authentication nor encryption, an attacker can access the status information for the drones that are equipped with telemetry transceivers. However, she cannot track the drones because the information has no identity that can bind the identities of the drones with their locations.

In this article, we proposed a fingerprinting method that uses the offsets of MEMS gyroscopes to identify and track the drones. Based on our results and through open-source drone firmware analysis, we found that the offsets obtained via the telemetry are a scaled version of the offsets of the raw outputs of MEMS gyroscopes. Moreover, because the offsets are calculated during the boot process and not changed during flight, the attacker can identify the flying drones. Experimental results using 70 stand-alone MEMS gyroscopes show that the proposed method achieves an F-score of 98.78% for identifying MEMS gyroscopes using 128 raw output samples of all three axes and an F-score of 94.47% using samples of two axes in the same setup. We also show that our fingerprints are effective, robust, and persistent despite the changes in temperature and time. In addition, the proposed method can be utilized to track commercial drones for drone traffic management systems, such as UTM.

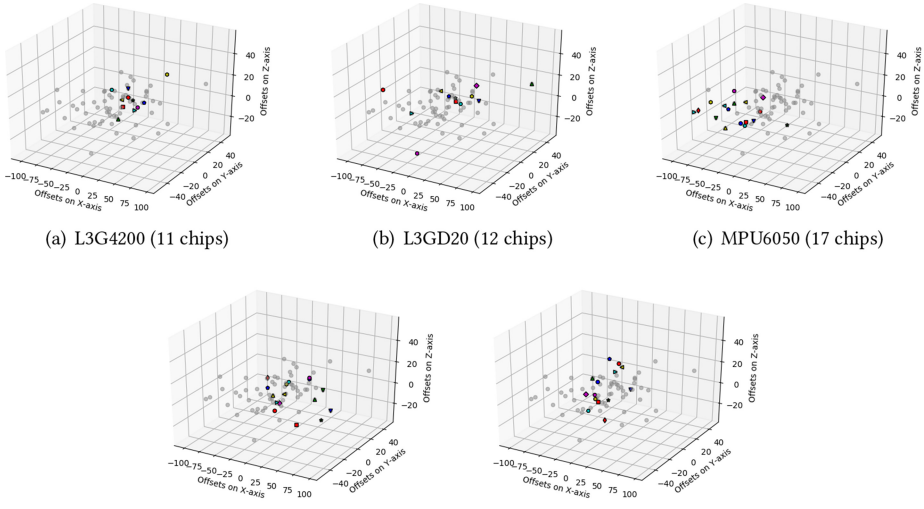
In the future, commercial drones may provide their identifying codes in public [45]. Because these identifiers can be maliciously used from an attacker's viewpoint, developers and researchers need to consider both security and safety issues to broadcast them.

APPENDIX

A INTRA-DISTANCE DISTRIBUTION

In Section 4, the spatial distribution of the offsets (Figure 7) and the cumulative distribution of the Euclidean distances between two offsets (Figure 8) are illustrated for all data that we collected. Both figures show the distributions of inter-distance among the fingerprints from 70 MEMS gyroscopes. To compare the characteristics of the collected fingerprints for each gyroscope model, the distributions of intra-distance are illustrated in this appendix.

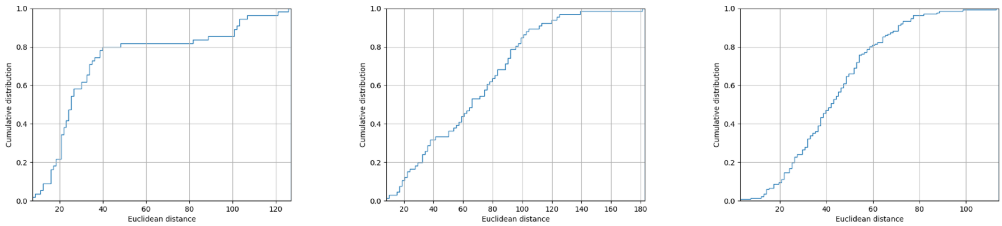
Figure 13 visualizes the spatial distribution of the fingerprints for each gyroscope model. We can observe that most of the fingerprints collected from one gyroscope model are grouped in different spaces. The cumulative distributions of the Euclidean distances for each gyroscope model are shown in Figure 14. The higher slope of the cumulative distribution graph means the more spatially scattered, and thus the easier to classify the fingerprints. In our experiments, the fingerprints of L3G4200 are the most spread among five gyroscope models.



(a) L3G4200 (11 chips) (b) L3GD20 (12 chips) (c) MPU6050 (17 chips)

(d) MPU6500 (16 chips) (e) MPU9150 (14 chips)

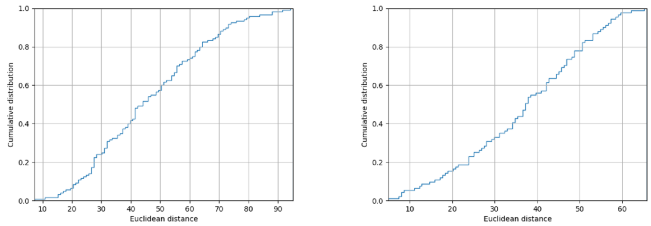
Fig. 13. Spatial distribution of the offsets for three axes per each gyroscope model. (Each point is averaged for all samples per axis and represented one MEMS gyroscope. The gray points are the samples of others.)



(a) L3G4200 (11 chips, the min. and max. distances are 7.54 and 127.08)

(b) L3GD20 (12 chips, the min. and max. distances are 8.13 and 182.89)

(c) MPU6050 (17 chips, the min. and max. distances are 2.94 and 114.08)



(d) MPU6500 (16 chips, the min. and max. distances are 7.25 and 95.12)

(e) MPU9150 (14 chips, the min. and max. distances are 4.97 and 65.78)

Fig. 14. A cumulative distribution of the Euclidean distances between two fingerprints per each gyroscope model.

However, this difference is not enough to identify gyroscope models because the spaces overlap each other. We note that our main purpose is to classify each gyroscope using its fingerprint, not each gyroscope model.

REFERENCES

[1] 3DR Solo—The Smart Drone. 2015. Retrieved from https://3dr.com/solo-drone.

[2] Chrisil Arackaparambil, Sergey Bratus, Anna Shubina, and David Kotz. 2010. On the reliability of wireless fingerprinting using clock skews. In *Proceedings of the Third ACM Conference on Wireless Network Security*. ACM, 169–174.

[3] Adam J. Aviv, Benjamin Sapp, Matt Blaze, and Jonathan M. Smith. 2012. Practicality of accelerometer side channels on smartphones. In *Proceedings of the 28th Annual Computer Security Applications Conference*. ACM, 41–50.

[4] Hristo Bojinov, Yan Michalevsky, Gabi Nakibly, and Dan Boneh. 2014. Mobile device identification via sensor fingerprinting. *arXiv:1408.1416* (2014).

[5] Liang Cai and Hao Chen. 2011. TouchLogger: Inferring keystrokes on touch screen from smartphone motion. In *Proceedings of the 6th USENIX Conference on Hot Topics in Security (HotSec'11)*.

[6] Liang Cai and Hao Chen. 2012. On the practicality of motion based keystroke inference attack. In *Trust and Trustworthy Computing*. Springer, Berlin.

[7] Marco Caselli, Dina Hadžiosmanović, Emmanuele Zambon, and Frank Kargl. 2013. On the feasibility of device fingerprinting in industrial control systems. In *International Workshop on Critical Information Infrastructures Security*. Springer, 155–166.

[8] Anupam Das, Nikita Borisov, and Matthew Caesar. 2014. Do you hear what I hear? Fingerprinting smart devices through embedded acoustic components. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.

[9] Anupam Das, Nikita Borisov, and Matthew Caesar. 2016. Tracking mobile web users through motion sensors: Attacks and defenses. In *Proceedings of the 23rd Annual Network and Distributed System Security Symposium (NDSS'16)*.

[10] Sanorita Dey, Nirupam Roy, Wenyuan Xu, Romit Roy Choudhury, and Srihari Nelakuditi. 2014. AccelPrint: Imperfections of accelerometers make smartphones trackable. In *Proceedings of the 21rd Annual Network and Distributed System Security Symposium (NDSS'14)*.

[11] Kevin Finisterre. 2015. Shelling out on 3DR Solo. Retrieved from http://www.digitalmunition.com/ShellingOutOnSolo_nopass.pdf.

[12] Ryan M. Gerdes, Thomas E. Daniels, Mani Mina, and Steve Russell. 2006. Device identification via analog signal fingerprinting: A matched filter approach. In *Proceedings of the 13rd Annual Network and Distributed System Security Symposium (NDSS'06)*.

[13] Github. 2010. ArduPilot Project. https://github.com/ArduPilot/ardupilot.

[14] Github. 2011. PX4 Firmware for ArduPilot. https://github.com/ArduPilot/PX4Firmware.

[15] Jun Han, Emmanuel Owusu, Le T. Nguyen, Adrian Perrig, and Joy Zhang. 2012. Accomplice: Location inference using accelerometers on smartphones. In *Proceedings of the IEEE International Conference on Communication Systems and Networks*.

[16] Craig Hillman and Cheryl Tulkoff. 2009. Manufacturing and reliability challenges with QFN. *SMTA DC Chapter* (2009).

[17] Bo-Jhang Ho, Paul Martin, Prashanth Swaminathan, and Mani Srivastava. 2015. From pressure to path: Barometer-based vehicle tracking. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. ACM, 65–74.

[18] iFixit. 2010. An example of MEMS gyroscope structure. Retrieved from https://www.ifixit.com/Teardown/iPhone+4+Gyroscope+Teardown/3156.

[19] InvenSense. 2013. InvenSense MPU6000/6050 datasheet. Retrieved from https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf.

[20] InvenSense. 2013. InvenSense MPU9150 datasheet. Retrieved from https://www.invensense.com/wp-content/uploads/2015/02/MPU-9150-Datasheet.pdf.

[21] InvenSense. 2014. InvenSense MPU6500 datasheet. Retrieved from https://www.invensense.com/wp-content/uploads/2015/02/MPU-6500-Datasheet2.pdf.

[22] Sitaraman Iyer and Tamal Mukherjee. 2002. Simulation of manufacturing variations in a Z-axis CMOS-MEMS gyroscope. In *Proceedings of the International Conference on Modeling and Simulation of Microsystems*, Vol. 1. Citeseer, 186–189.

[23] Samy Kamkar. 2013. SkyJack. http://samy.pl/skyjack/.

[24] Tadayoshi Kohno, Andre Broido, and Kimberly C. Claffy. 2005. Remote physical device fingerprinting. *IEEE Trans. Dependable Secure Comput.* 2, 2 (2005), 93–108.

[25] Xiang-Yang Li and Zhiguang Qin. 2015. Wireless device authentication using acoustic hardware fingerprints. In *Proceedings of the Big Data Computing and Communications: First International Conference (BigCom'15), Taiyuan, China, August 1-3, 2015*, Vol. 9196. Springer, 193.

[26] Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. 2011. (sp) iPhone: Decoding vibrations from nearby keyboards using mobile phone accelerometers. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*. ACM, 551–562.

[27] Joseph A. Marty. 2013. *Vulnerability Analysis of the MAVLink Protocol for Command and Control of Unmanned Aircraft (No. AFIT-ENG-14-M-50)*. Master's thesis. Air Force Institute of Technology.

[28] mavlink.org 2009. MAVLink: Micro Air Vehicle Communication Protocol. http://mavlink.org/messages/common, http://qgroundcontrol.org/mavlink/start, and https://en.wikipedia.org/wiki/MAVLink.

[29] Andryas Mawardi and Ranga Pitchumani. 2005. Design of microresonators under uncertainty. *J. Microelectromechanical Syst.* 14, 1 (2005), 63–69.

[30] Maryam Mehrnezhad, Ehsan Toreini, Siamak Shahandashti, and Feng Hao. 2016. Stealing PINs via mobile sensors: Actual risk versus user perception. In *Proceedings of the 1st European Workshop on Usable Security (EuroUSEC'16)*.

[31] Maryam Mehrnezhad, Ehsan Toreini, Siamak F. Shahandashti, and Feng Hao. 2016. TouchSignatures: Identification of user touch actions and PINs based on mobile sensor data via javascript. *J. Inform. Secur. Appl.* 26 (2016), 23–38.

[32] Yan Michalevsky, Dan Boneh, and Gabi Nakibly. 2014. Gyrophone: Recognizing speech from gyroscope signals. In *23rd USENIX Security Symposium (USENIX Security'14)*. 1053–1067.

[33] Emiliano Miluzzo, Alexander Varshavsky, Suhrid Balakrishnan, and Romit Roy Choudhury. 2012. Tapprints: Your finger taps have fingerprints. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*. ACM, 323–336.

[34] Sashank Narain, Amirali Sanatinia, and Guevara Noubir. 2014. Single-stroke language-agnostic keylogging using stereo-microphones and domain specific machine learning. In *Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks*. ACM, 201–212.

[35] Sashank Narain, Triet D. Vo-Huu, Kenneth Block, and Guevara Noubir. 2016. Inferring user routes and locations using zero-permission mobile sensors. In *Proceedings of the IEEE Symposium on Security and Privacy*.

[36] Emmanuel Owusu, Jun Han, Sauvik Das, Adrian Perrig, and Joy Zhang. 2012. Accessory: Password inference using accelerometers on smartphones. In *Proceedings of the ACM Workshop on Mobile Computing Systems & Applications*.

[37] Pixhwak1. 2010. Pixhwak1 Flight Controller. Retrieved from https://pixhawk.org/modules/pixhawk.

[38] Pixhwak2. 2016. Pixhwak2 Flight Controller. Retrieved from http://www.proficnc.com.

[39] QGroundControl. 2010. QGroundControl. Retrieved from http://qgroundcontrol.com.

[40] Kurt Rosenfeld, Efstratios Gavas, and Ramesh Karri. 2010. Sensor physical unclonable functions. In *Proceedings of the 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'10)*. IEEE, 112–117.

[41] F. Samland, J. Fruth, M. Hildebrandt, T. Hoppe, and J. Dittmann. 2012. AR.drone: Security threat analysis and exemplary attack to track persons. In *Proceedings of the Society of Photo-Optical Instrumentation Engineers Conference Series*.

[42] Ryan A. Scheel and Akhilesh Tyagi. 2015. Characterizing composite user-device touchscreen physical unclonable functions (PUFs) for mobile device authentication. In *Proceedings of the 5th International Workshop on Trustworthy Embedded Devices*. ACM, 3–13.

[43] Shellntel. 2015. Drone Code Execution. Retrieved from http://www.shellntel.com/blog/2015/9/25/drone-code-execution.

[44] Laurent Simon and Ross Anderson. 2013. Pin skimmer: Inferring pins through the camera and microphone. In *Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*. ACM, 67–78.

[45] IEEE Spectrum. 2017. Electronic license plates for drones. Retrieved from https://spectrum.ieee.org/automaton/robotics/drones/electronic-license-plates-for-drones.

[46] Raphael Spreitzer. 2014. Pin skimming: Exploiting the ambient-light sensor in mobile devices. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*. ACM, 51–62.

[47] STMicroelectronics. 2010. STMicroelectronics L3G4200D datasheet. Retrieved from http://www.st.com/resource/en/datasheet/l3g4200d.pdf.

[48] STMicroelectronics. 2013. STMicroelectronics L3GD20 datasheet. Retrieved from http://www.st.com/resource/en/datasheet/l3gd20.pdf.

[49] jDrones. 2014. Longrange Telemetry Set (jD-RF900Plus). Retrieved from http://store.jdrones.com/jD_RD900Plus_Telemetry_Bundle_p/rf900set02.htm.

[50] NASA UTM. 2015. Google UAS airspace system overview. Retrieved from https://utm.arc.nasa.gov/docs/GoogleUASAirspaceSystemOverview5pager[1].pdf.

[51] He Wang, Ted Tsung-Te Lai, and Romit Roy Choudhury. 2015. Mole: Motion leaks through smartwatch sensors. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 155–166.

[52] Oliver Willers, Christopher Huth, Jorge Guajardo, and Helmut Seidel. 2016. MEMS gyroscopes as physical unclonable functions. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 591–602.

[53] NASA. 2015. First steps toward drone traffic management. Retrieved from http://www.nasa.gov/feature/ames/first-steps-toward-drone-traffic-management.

[54] NASA. 2015. UTM: Air traffic management for low-altitude drones. Retrieved from http://www.nasa.gov/sites/default/files/atoms/files/utm-factsheet-11-05-15.pdf.

[55] sUAS News. 2015. Market share by FAA registration. Retrieved from http://www.suasnews.com/2015/06/forty-eight-percent-of-commercial-drone-platforms-in-the-usa-made-by-dji/.

[56] Zhi Xu, Kun Bai, and Sencun Zhu. 2012. Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. In *Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, 113–124.

[57] Zhe Zhou, Wenrui Diao, Xiangyu Liu, and Kehuan Zhang. 2014. Acoustic fingerprinting revisited: Generate stable device ID stealthily with inaudible sound. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 429–440.

Received March 2017; revised December 2017; accepted December 2017