Intrusion Detection Systems-Enabled Power Electronics for Unmanned Aerial Vehicles

Mohammad Ashiqur Rahman*, Md Tauhidur Rahman†, Mithat Kisacikoglu‡, Kemal Akkaya*
*Department of Electrical and Computer Engineering, Florida International University, Miami, FL, USA
†Department of Electrical and Computer Engineering, University of Alabama in Huntsville, AL, USA
‡Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa, AL, USA
Emails: marahman@fiu.edu, tauhidur.rahman@uah.edu, mkisacik@ua.edu, kakkaya@fiu.edu

Abstract—Compromised power electronics, due to firmware attacks and hardware Trojans, in a flight computer can jeopardize the safety and security of an Unmanned Aerial Vehicle (UAV). They can maliciously alter sensor measurements or control commands to make a UAV to take disastrous moves. Unfortunately, most of these attacks are difficult to detect before deploying components in the system. Therefore, detecting compromised behavior run-time is important, while it is challenging at the same time. In this work, we propose to build machine learning-based intrusion detection systems (IDSs) to be deployed at the power electronics/microcontroller level such that it can deal with malicious data/control commands initiated due to hardware attacks.

Index Terms—Unmanned aerial vehicles; hardware attacks; power electronics; intrusion detection; machine learning.

I. Introduction

Unmanned Aerial Vehicles (UAVs) or drones are increasingly being used for remote surveillance as well as data collection. They are successfully used in scenarios where it is highly dangerous, as well as monotonous for human observers, or in some cases, excessively costly. For example, UAVs play an important role in security and safety-critical applications, such as surveillance of (or collecting data at) sensitive points in industrial networks, nuclear systems, transportation, and international borders [1]–[5]. They can also be used in UAV-assisted wireless sensor networks to collect data directly from sensor nodes [6]. To complete the assigned tasks, a UAV needs to collect and process sensitive data. However, malicious modification of such data and the controller's operation can jeopardize the safety and security of the UAV.

Many real attacks on UAVs have been reported. The video feeds from fixed-wing UAVs were intercepted by launching attacks by Iran on American drones in December 2009, which was identified later when U.S. military personnel in Iraq found the video feeds on the apprehended militant's laptop [7]. Another attack incident was reported later in 2012, where a fixed-wing UAV was suspected to be captured by Iran [8], [9]. In May 2012, a GPS jamming attack was suspected to be executed on a UAV that caused it to crash on the ground control van killing an engineer and injuring two others during testing [10]. Similar to firmware, a hardware Trojan can also manipulate the operation of a controller. For example, a hardware Trojan in a pulse-width modulation (PWM) controller can manipulate the operation and help the attacker take control of the UAV.

Given the growing number of hardware and firmware based attacks on UAVs, securing them against malicious activities are of utmost importance. Otherwise, malfunctioning of mission- and safety-critical applications can cause serious injury and damage at various levels.

In this paper, we briefly discuss the role of firmware and hardware Trojan based attacks that mainly manipulate or change data/command maliciously. We also develop machine-learning-based intrusion detection systems (IDSs) at the power electronics/microcontorller level to prevent UAV system from the consequences of firmware-based or hardware Trojan based attacks. The study is limited into a preliminary discussion about the approach and corresponding implementation challenges.

This paper is organized as follows. Section II overviews the flight controller components of a UAV. Section III briefly discusses hardware Trojan and firmware-based attacks that can be used to jeopardize the safety and security of the UAV system. In the following section, intrusion detection-based countermeasures against malicious modification of data/command is proposed, which is followed by the conclusion.

II. Flight Controller Overview

There are different firmware programs that contain necessary tools for controlling different types of UAVs. PX4 is a popular open flight-control software for drones and other UAVs [11] that includes drivers for different types of sensors, sensor fusion algorithms, feedback control algorithms, data-logging and communication applications, generation of control signal for different actuators (i.e., power electronic circuits for motors), and predefined operation state machines. The PX4 firmware runs on a properly configured operating system (e.g., Nuttx [12]) installed on a supported hardware environment – a flight computer (e.g., PixHawk [13]). Fig. 1 shows the workflow structure of a flight controller.

The controller gets inputs from various sensors, e.g., magnetometer, accelerometer, gyroscope, barometer, GPS, etc., most of which are on-board. These measurements can pass through an Extended Kalman Filter [14]

978-1-7281-9591-9/20/$31.00 ©2020 IEEE

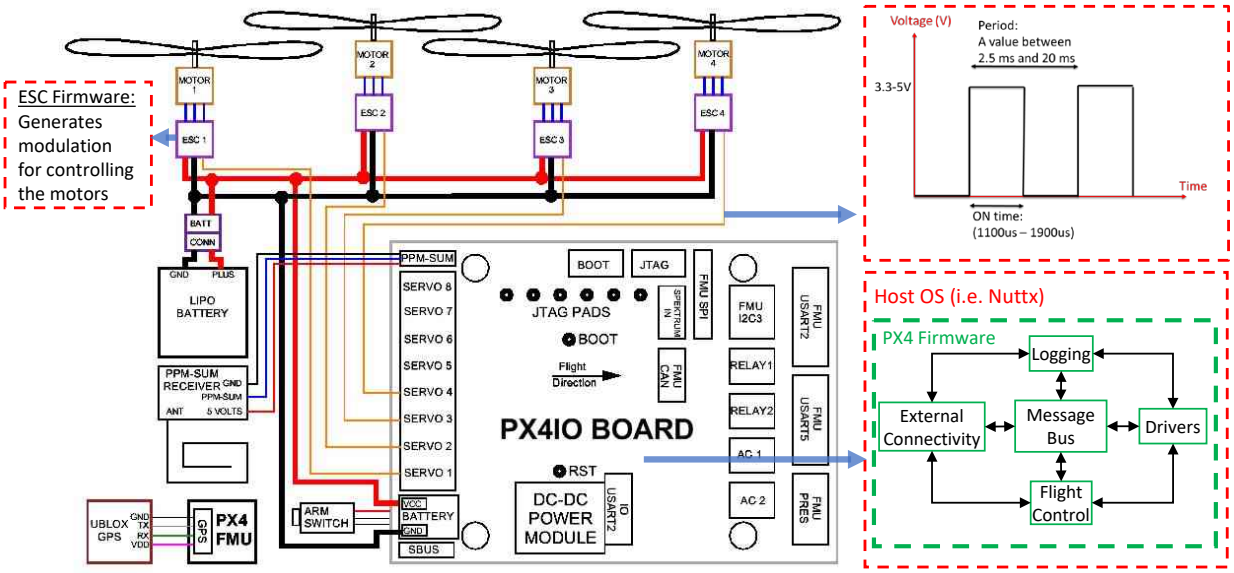

Fig. 1. A typical workflow structure of the flight controller (PX4 firmware) on a UAV/drone hardware.

to reliably estimate different input parameters for a series of control submodules, namely position control, speed control, orientation control, and angular rates control. The control structure also takes remote pilot/orientation commands as well as trajectory paths (waypoints) as inputs. The angular rate control submodule finally sends the control commands to the inverters at the multi-copter motors as pulse width modulation (PWM) signals. Note that PWM does not stand here for inverter switch gate driver signals but a low-frequency reference signal for the inverter to track.

In the PX4, a group of applications performs specific tasks for allowing a vehicle to operate, and different parts of the board work together using the publisher-subscriber protocol. As part of the initialization code of the publisher application, the application needs to register itself as the publisher of a specific message. Applications that require the information provided by a message, register themselves as subscribers of that message.

III. Potential Attacks on a UAV Flight Controller

Various potential attacks can be used to take control of a flight controller. Below, we have summarized a few of the potential attacks.

A. False Data Injections to Sensor Data

The data generated from the sensors are used for the flight control system to analyze and generate control commands. However, an attacker can inject faults in the sensor data to cause erroneous commands and undesirable consequences. Theses faults can be injected in (i) one or (ii) multiple sensors/measurements (e.g., magnetometer, accelerometer, gyroscope, barometer, GPS, etc.). False injection can be categorized into two major types: (i) random injection where the attackers aim at injecting false data randomly and (ii) stealthy injection. The idea of stealthy injection is to keep a measurement change harmonious with other measurements.

To understand the potential impacts of the attacks on the system, we must have the understanding of the following points: (i) the list of data items (measurements) that can be falsified; (ii) the control routines that use the data, where one routine often needs multiple sensor measurements while multiple routines use the measurements from the same sensor ; and (iii) algorithms or logic of control routines.

B. Firmware Attacks

Malicious attackers can insert malware in the system's firmware in order to steal sensitive information or take control of the whole controller. There are different types of potential firmware attacks, including the following:

- Maliciously crafted input: In this method, an attacker might use buffer overflows to inject malware.
- Elevation of privilege: An attacker circumvents security functions through System Management Mode (SMM) code injection.
- Data tampering: An attacker modifies UEFI variables (SecureBoot, Configuration, etc.).
- Unauthorized access to sensitive data: Disclosure of System Management Random Access Memory (SMRAM) contents.
- Information disclosure: SMM rooted malware; "secrets" left in memory.
- Denial of Service: Serial peripheral interface (SPI) flash corruption to "brick" the system.

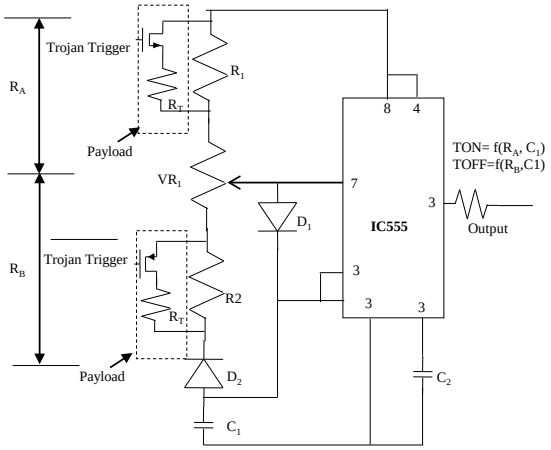

Fig. 2. PWM generation circuit and potential Trojans.

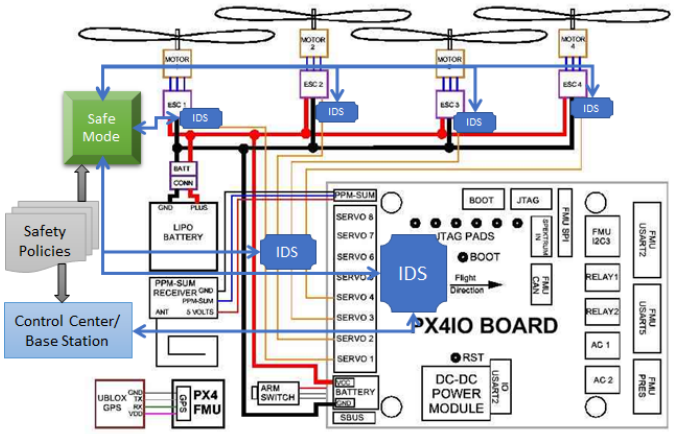

Fig. 3. The deployment of IDSs on the UAV hardware.

C. Hardware Trojans

In the modern semiconductor supply chain, a hardware Trojan can be inserted by an untrusted party [?], [15], [16]. These Trojans are capable of leaking sensitive information, disabling key portions of the IC, self-destructing the chip, or hindering performance [?]. PWM is a digital signal which is used in UAV control circuitry. The PWM generated signal remains high for a certain amount of time, which is known as on-time. On the other hand, the time period when the signal remains low is known as off time. The on-time and off-time of a signal control the speed of a motor. However, the on-time and off-time of a PWM signal can be maliciously modified by an attacker in the untrusted supply chain to result in undesired behavior when the Trojan inserted PWM generator is deployed in an electronic system. Different components of the PWM generator can be targeted to insert a hardware Trojan. Fig. 2 shows a simplified PWM generator circuit, where resistors R_A and R_B and capacitor C_1 are tuned to change on- and off-time of a PWM signal. An attacker can design a Trojan trigger that monitors various signals and/or a series of events in the circuit/system (not shown in Fig. 2). Once the trigger detects an expected event or condition, the payload (shown in Fig. 2) is activated to perform malicious behavior. When Trojan activated, R_A will deviate from the expected R_A to cause unexpected behavior.

IV. UAV Power Electronics Defense

In this paper, we propose to develop intrusion detection systems (IDSs) to detect tampered data/commands to/from power electronics. These data are usually input or output of different components (sensors, microcontrollers, and actuators) of the device control system. We propose IDSs to be deployed in three layers (Fig. 3): (i) at the controller level (e.g., the flight computer), (ii) on the communication channel between the controller and the inverters, and (iii) at the individual inverter level.

A. Learning with Normal Data

One of the purposes of pulse width modulation (PWM) in the electrical system is to control the speed of the motors. The ON and OFF time of the signal is modulated to control the average value of the signal. The on-time contains the information of the commanded speed to the motor, and the mapping between the on-time and the desired speed of the motor is usually linear. We will train the model based on the normal data with respect to PWM signals, collected in the following three categories.

- Relationship among different PWM signals: There are often several inverters that control speed/torque of electric motors on the system. The data set will consider the PWM signals from the output driver of the flight controller to various inverters, and the relationship between them will be learned.
- Relationship among the consecutive PWM signals: The data set will consider multiple consecutive PWM signals. The relationship between these consecutive PWM signals will be learned.
- Valid PWM signals to an individual inverter: The data set will consider subsequent PWM signals to an inverter to learn the expected signal patterns.

Learning based on the sensor data will follow a similar learning pattern as that of PWM signals.

B. Proposed Learning Idea with Attack Data

Along with the normal data, we will use the attack data to train the model. The attack data sets will be collected in the following directions:

- Learning the dissimilarity among different PWM signals: One or some PWM signals can be corrupted (i.e., false data injection) due to the hardware trojan (e.g., deployed at the output drive) or the compromised firmware. The data set of corrupted PWM signals to various inverters will be used to learn attack features.
- Dissimilarity based on the consecutive PWM signals: The data set will consider multiple consecutive PWM

signals, which includes the normal signals and the attack signals. The transition between normal to attack signal and the relationship between consecutive PWM corrupted signals will be learned.
- Learning dissimilarity based on the PWM signals to an individual inverter: The data set will consider consecutive PWM signals to an inverter. This data will be used to learn the expected values of the signal.

C. Proposed Learning Approach

The developed IDSs in this task will classify the received sensor data to the flight computers as well as the PWM signals sent to the inverters as normal or malicious. We consider three types of IDSs in our task.

1) Attack Detection at the Controller: For anomaly detection in the controller, we consider an IDS that can classify a control decision. We train our learning model using a two-class support vector machine (SVM) based machine learning model. It is worth mentioning why we prefer two-class SVM over one-class. The latter kind of machine learning models can be used to detect anomaly using positive data only by drawing a non-linear boundary around the positive data samples, which would eliminate the necessity of attack data in our training dataset. However, the hyperparameters need to be tuned to perfectly fit the positive data samples. A wrong estimation of this value raises the risk of overfitting or underfitting the model, compared to the two-class SVM model. Having attack data samples, although they will be much smaller in number compared to the normal ones, will help us reduce potential false alarms. Our dataset will include sensor measurements (magnetometer, accelerometer, gyroscope, barometer, GPS, etc.) and corresponding control decision (duty cycle for each motor) as features for both normal and attack cases, and check for an anomaly.

SVM Modeling. Let us assume that S is the set of sensor measurements and D is a set of duty cycles for operating and controlling motor speed. Depending on these values, our model assigns a label to the data to be either normal or attack data. Our two-class SVM model attempts to differentiate between two classes in the dataset by drawing a hyperplane with the intent to maximally divide both classes. The model can fit optimal decision boundary to separate both linear and nonlinear data. Initially, the system will be trained using both the normal and attack data sets X $(N \cup A)$. Afterward, the IDS will be deployed to verify real-time sensor data/PWM signals, and based on the trained SVM model it will detect whether the received data is representing an attack or not. Two-class classifier is inspired by the SVM classifier [17], [18]. Two-class classification problem finds a hyperplane that can separate the desired fraction of one type of training patterns from another type. SVM maps the input data to the feature dimension based on the kernel function.

Implementing SVM model. We will train the two-class SVM model offline for getting model parameters. After acquiring model parameters, we will use that model to build our IDS in the controller level for finding any abnormality produced by the discrepancy in the sensor measurements, or controller decision. The dataset considered in our case is most likely to be nonlinear. We will consider kernel SVM (e.g., Gaussian, polynomial, Sigmoid, or computable kernel) that attempts to linearly separate the non-homogeneous data points by projecting the data into high dimensional feature space.

2) Attack Detection at the PWM Converter: Due to some on-board malware or hardware Trojan, the generated control decision can be compromised before getting converted into the converter's PWM signal. Hence, we consider another IDS in the converter for boosting the security of the system. We think of an IDS to be implanted in the converter that undertakes a couple of techniques for figuring abnormality in the system. Before applying anomaly detection, we will preprocess the PWM converted signal using digital signal processing (DSP) and recover our duty cycle information. If there is any inconsistency of this information compared to the controller generated decision information, our proposed IDS will issue an alarm. Again, to add security in this IDS, we will utilize an SVM-based novelty detection technique.

3) Attack Detection at ESCs: The Control center generates PWM signals for each of the motors connected through the ESC. The PWM signals are transmitted through the communication lines between the control center and the ESCs. The ESCs trigger the inverter using the PWM signal to run the motors at the desired speed. However, as discussed earlier, the ESCs themselves can be compromised with a hardware Trojan. Trojan's target is to generate an abnormal output at the inverter, which may cause unexpected speed in the motors. Thus, a lightweight rule-based IDS will be installed in each ESC that will compare received PWM signals and the corresponding inverters' outputs. Voltage and current sensors are used to measure the output of the inverter. After processing both the signals coming to the ESC and generated by the inverter, the duty cycle of the signals will be calculated. An IDS rule will verify the equality of these two duty cycles for being normal.

V. Conclusion

This study focuses on investigating security of the flight controller-based power electronics in UAVs. We briefly discussed the potential firmware and hardware-based attacks at the UAV power electronics. We presented the idea of building IDSs to defend the flight control system against such on-board attacks. The proposed approach will apply appropriate techniques to build IDSs at three levels of the control and actuation path, starting from the flight controller down to the actuators. The

proposed detection techniques include sophisticated ML-based anomaly detection as well as simple rule-based matching depending on the computing capability at different layers. Our future work will include collecting the PWM data and corresponding control commands for UAV power electronics and training/building the IDSs.

References

[1] S. Srinivasan, H. Latchman, J. Shea, T. Wong, and J. McNair, "Airborne traffic surveillance systems: video surveillance of highway traffic," in ACM Workshop on Video Surveillance & Sensor Networks, 2004, pp. 131–135.

[2] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous uav surveillance in complex urban environments," in IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2009, pp. 82–85.

[3] C. Deng, S. Wang, Z. Huang, Z. Tan, and J. Liu, "Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications," J. Commun, vol. 9, no. 9, pp. 687–692, 2014.

[4] L. Li, "The uav intelligent inspection of transmission lines," in International Conference on Advances in Mechanical Engineering and Industrial Informatics, 2015, pp. 1542–1545.

[5] C. A. Trasviña-Moreno, R. Blasco, Á. Marco, R. Casas, and A. Trasviña-Castro, "Unmanned aerial vehicle based wireless sensor network for marine-coastal environment monitoring," Sensors, vol. 17, no. 3, p. 460, 2017.

[6] A. E. Abdulla, Z. M. Fadlullah, H. Nishiyama, N. Kato, F. Ono, and R. Miura, "An optimal data collection technique for improved utility in uas-aided networks," in INFOCOM, 2014 Proceedings IEEE. IEEE, 2014, pp. 736–744.

[7] "Iraq insurgents 'hack into video feeds from us drones'," December 2009. [Online]. Available: http://news.bbc.co.uk/2/hi/world/middle_east/8419147.stm

[8] "Iran says it captures drone; u.s. denies losing one," December 2012. [Online]. Available: https://www.reuters.com/article/us-iran-usa-drone/iran-says-it-captures-drone-u-s-denies-losing-one-idUSBRE8B308920121204

[9] L. He, W. Li, C. Guo, and R. Niu, "Civilian unmanned aerial vehicle vulnerability to gps spoofing attacks," in Seventh International Symposium on Computational Intelligence and Design, vol. 2, 2014, pp. 212–215.

[10] Y.-S. Lee, Y.-J. Kang, S.-G. Lee, H. Lee, and Y. Ryu, "An overview of unmanned aerial vehicle: Cyber security perspective," 08 2016, pp. 128–131.

[11] "Getting started with px4." [Online]. Available: https://px4.io/documentation/

[12] "Nuttx, a real-time embedded operating system (rtos)." [Online]. Available: https://bitbucket.org/nuttx/nuttx/src/master/

[13] "Pixhawk, an independent open-hardware standards for autopilots." [Online]. Available: https://pixhawk.org/

[14] D. Simon, Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches. USA: Wiley-Interscience, 2006.

[15] M. T. Rahman, D. Forte, Q. Shi, G. K. Contreras, and M. Tehranipoor, "Csst: Preventing distribution of unlicensed and rejected ics by untrusted foundry and assembly," in 2014 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2014, pp. 46–51.

[16] K. Worley and M. T. Rahman, "Supervised machine learning techniques for trojan detection with ring oscillator network," in 2019 SoutheastCon, 2019, pp. 1–7.

[17] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," in the 12th International Conference on Neural Information Processing Systems. Cambridge, MA, USA: MIT Press, 1999, pp. 582–588.

[18] V. N. Vapnik and V. Vapnik, Statistical learning theory. Wiley Press, New York, 1998, vol. 1.