This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2863237, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Empirical Analysis of MAVLink Protocol Vulnerability for Unmanned Aerial Vehicles

YOUNG-MIN KWON, JAEMIN YU, BYEONG-MOON CHO, YONGSOON EUN, (MEMBER, IEEE) AND KYUNG-JOON PARK, (Member, IEEE)
Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu 42988, South Korea

Corresponding author: Kyung-Joon Park (e-mail: kjp@dgist.ac.kr)

This work was partly supported by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (2014-0-00065, Resilient Cyber-Physical Systems Research), the DGIST R&D Program of the Ministry of Science and ICT(18-ST-02). and Unmanned Vehicles Advanced Core Technology Research and Development Program Through the Unmanned Vehicle Advanced Research Center (UVARC) funded by the Ministry of Science, ICT and Future Planning, the Republic of Korea (NRF-2016M1B3A1A01937599).

ABSTRACT Recently, unmanned aerial vehicle (UAV) or so called drones are used in various applications. Especially, UAVs are used for rescue systems, disaster detection, and military purposes, as well as for leisure and commercial purposes. In particular, UAVs that are controlled over networks by ground control stations (GCS) can provide various services with the expanded activity area. Consequently, it is of critical importance to investigate the vulnerability of the drone system. In this paper, we focus on the UAVs controlled by GCS over networks. We analyze the vulnerability of the micro air vehicle communication (MAVLink) protocol, which is one of the most widely adopted communication protocols for GCS-based control of UAVs. Then, by exploiting the vulnerability of the MAVLink protocol, we propose an attack methodology that can disable an ongoing mission of a UAV. Our empirical study confirms that the proposed attack can stop the attacked UAV and disable the mission.

INDEX TERMS UAV, UAS, drones, MAVLink, network attack, DoS, flooding attack, packet injection

I. INTRODUCTION

RECENTLY, unmanned aerial vehicles (UAV), or so called drones, have been widely used around the world for the last decade. Especially, we can think of various services by drones. For example, drone network provides services for various drone applications [1], such as rescue systems [2], disaster monitoring [3, 4], commercial use, military mission, and so on.

An example of a commercial service using UAVs is Amazon's project Prime-Air, which was released in 2015 [5]. This system aims to design a future delivery service using UAVs. Since then, various services utilizing UAVs such as Fleetlight [6] and Matternet [7] have been released, as shown in Fig. 1. In this way, services using UAVs are mainly performed in environments that are controlled over networks as shown in Fig. 2. Controlling the UAV over a network allows the UAV to perform its mission by completing the mission without user control.

However, as the demands for the services using UAVs are increasing, the negative use cases are also rapidly increasing. Therefore, we need methodologies that can disable malicious UAVs.

In this paper, we focus on the UAVs controlled by GCS over networks. We empirically analyze the vulnerability of the micro air vehicle communication (MAVLink) protocol, which is one of the most popular protocols used for GCS-based control of UAV [8]. It should be noted that there exist few empirical studies on the vulnerability of the MAVLink protocol. By exploiting the vulnerability of the MAVLink protocol, we propose an attack methodology that can disable an ongoing mission of the UAV. We empirically validate the proposed attack methodology with a UAV testbed. Our experimental results confirms that the attacked UAV is stopped and the mission disabled.

Our contributions can be summarized as follows:

- We identify the vulnerability of the MAVLink protocol, a de facto standard for UAV and GCS communication.
- By exploiting the identified protocol vulnerability, we

develop an attack methodology that can disable the mission of UAVs.

The rest of the paper is organized as follows. In Section II, we provide background information on drone controls, the MAVLink protocol, and network attack methods. In Section III, we introduce the proposed method to disable a UAV. The experimental environment and the experiment scenarios are presented in Section IV. In Section V, we summarize existing work on disabling UAVs. Finally, Section VI concludes this paper.

II. BACKGROUNDS

Here, we provide background materials for our study.

A. DRONE CONTROL STRUCTURE

There are basically two ways to control a UAV; using a controller and using a ground control station (GCS) as shown in Fig. 3. In controller-based control, the user views the UAV directly or watches through a camera mounted on the UAV and controls it using the controller. The UAV and the controller are connected to a communication module, and the UAV is controlled by transmitting the controller's signal to the UAV in real time. Generally, the communication modules used are telemetry, Wi-Fi, ZigBee, and so on.

On the other hand, GCS-based control uses a computer to connect the managing software and the UAV; GCS then performs mission commands uploaded by the user. GCS can monitor the status of the UAV by receiving information of various sensors mounted on the UAV such as current altitude, speed, map position, and current mission status. The controller-based method can manually control the UAV in real time while GCS-based control enables stable flight as well as unassisted flight to complete autonomous missions. Here, we consider GCS-based control for our study.

B. MAVLINK PROTOCOL

Here, we focus on the MAVLink protocol, which is one of the most widely used protocols for GCS-based drone control. The MAVLink protocol is a message-based UAV communication protocol developed by Lorenz Meier in 2009 [8]. In addition, the MAVLink protocol is a part of the current DroneCode project and is used by thousands of developers. It is also used in numerous Autopilot-based systems such as ArdupilotMega, pxIMU Autopilot, and SLUGS Autopilot [9]. MAVLink packets are bidirectionally transferred between UAV and GCS as header-based messages. The GCS sends mission commands to the UAV, and the UAV transmits state information including the sensor value, and current position to the GCS. Fig. 4 shows the message structure of the MAVLink protocol and Table 1 shows the meaning of the MAVLink frame [8].

TABLE 1. Meaning of the MAVLink frame [8].

Byte Index	Content	Value	Explanation
0	Packet Start Sign (STX)	0xFE	Indicates start of a new packet
1	Payload Length (LEN)	0-255	Indicates length of the following payload
2	Packet sequence (SEQ)	0-255	Packet transfer sequence information for detecting packet loss
3	System ID (SYS)	1-255	ID of the sending system; Allows to identify multiple platforms on the same network
4	Component ID (COMP)	0-255	ID of the sending component; Allows to identify multiple components on the same platform
5	Message ID (MSG)	0-255	ID of the message; Define what payload means, and how to decode it
6 to (n+6)	Data (Payload)	0-255 (bytes)	Data of message; depends on the message ID
(n+7) to (n+8)	Checksum (CKA and CKB)	ITU X.25/SAE AS-4 hash of bytes 1 to (n+6); It includes MAVLINK_CRC_EXTRA parameter computed from message fields	

C. NETWORK ATTACK

Network attacks violate the confidentiality, integrity, and availability of the system. Confidentiality allows information on the system only to authorized users. If confidentiality is violated, it is possible to eavesdrop on information and spoof the system. Integrity means the original information and signals transmitted, stored, and converted are maintained and not changed afterwards. Violation of integrity allows attacks such as message injection, replay attack, and so on. Availability allows the system to function for the time required by the user. In terms of maintenance, service must not be interrupted; performance must be maintained. Also, in terms of access to the system, the service must be accessible whenever the user needs it. Denial of service attacks can violate availability.

1) Man-In-The-Middle

Man-in-the-middle (MITM) is an attack that violates the confidentiality or integrity of the system [10, 11]. As can be seen from the name, the attacker is located in the middle of the hosts and sniffs information [12]. The attacker can cause hosts to communicate information to the attacker. This is possible because system allows host to set the destination address to the attacker's address for address resolution protocol (ARP) poisoning. When MITM is applied to the UAV system, it is possible to eavesdrop on all of information transmitted between the UAV and GCS.

2) Eavesdropping

Eavesdropping is an attack that violates the confidentiality of the system; it means that an attacker steals and listens to information of other users. If an MITM attack succeeds, eavesdropping can be enabled [12]. As a method to protect the system from eavesdropping, it is necessary to encrypt the message.

FIGURE 1. Fleetlight and Matternet service.

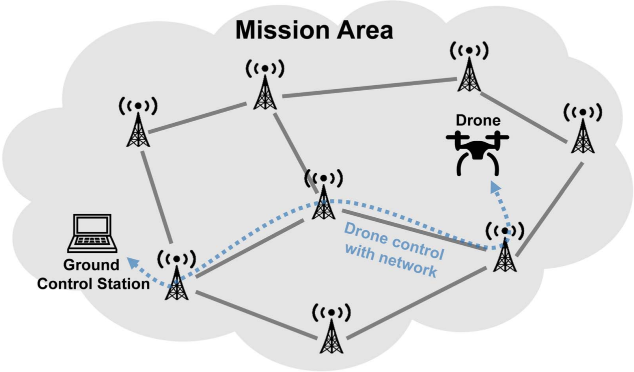


FIGURE 2. UAV system controlled over network.

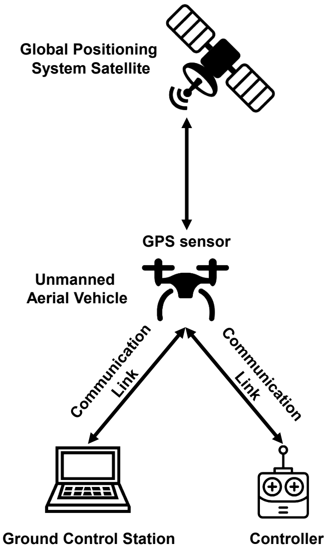


FIGURE 3. Two ways of drone control: GCS vs. direct controller.

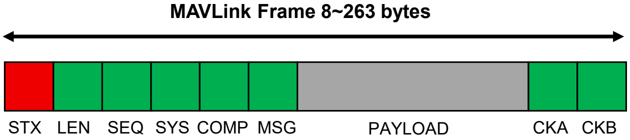


FIGURE 4. MAVLink protocol data frame structure [8].

3) Denial-of-Service

Denial-of-Service (DoS) attacks violate availability, monopolizing the resources of the system; using both DoS and MITM, it is possible to prevent other users from using system services [13]. In case of a DoS attack on a UAV system, control message, sensor information, and mission information are not correctly transmitted. Therefore, not only is the UAV not maintained in the stable state, but also the mission execution can not be performed correctly.

4) Potential threats to UAV systems

In the UAV system, it is possible to have different vulnerabilities for each component of the system. Therefore, the potential threats that may occur for each component may differ. The threats that can occur for each component of the UAV system are classified by the security objective [14, 15, 16, 17]. Table 2 shows the potential threats that may occur for each component of the UAV system.

III. PROPOSED ATTACK METHODOLOGY

A. VULNERABILITY OF THE MAVLINK PROTOCOL

Since the MAVLink message is a header-based protocol, it checks the first frame of the data packet and classifies the message. Therefore, it checks the STX value which is the initial frame and recognizes whether it is a MAVLink packet. In order to improve transfer speed and efficiency, the MAVLink message does not perform encryption [8]. When a message is encrypted, because the value of the header of the packet changes, the system does not recognize it as a MAVLink packet. Furthermore, it takes additional time to decrypt the data. Hence, the MAVLink protocol does not introduce encryption. Therefore, there exists a security

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2863237, IEEE Access

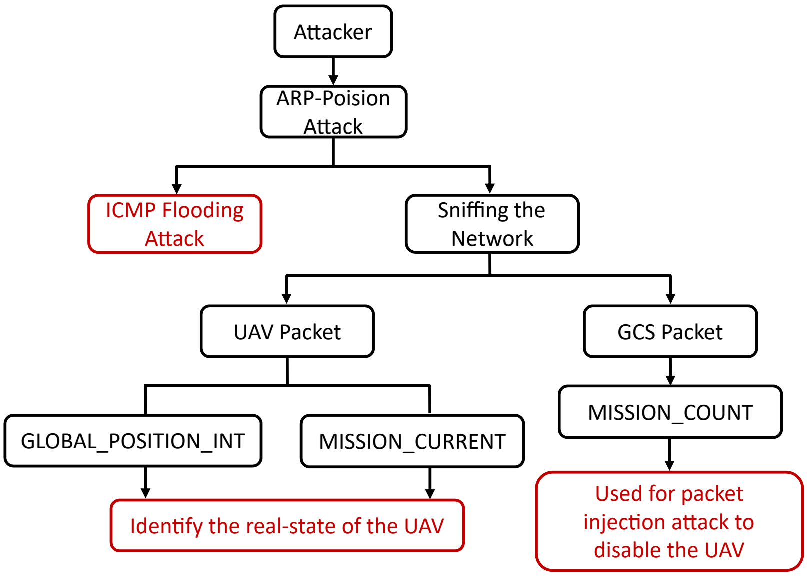


FIGURE 5. Overall procedure of UAV attack scenarios.

TABLE 2. Potential threats on UAV systems.

Security objective	System objective	Attack method
Confidentiality	GCS	Virus
		Malware
		Keyloggers
		Trojans
	UAV	Hijacking
	Communication Link	Eavesdropping
		Man-in-the-middle
Integrity	Communication Link	Packet injection
		Replay attack
		Man-in-the-middle
		Message deletion
Availability	GCS	Denial of service
	UAV	Fuzzing
	Communication Link	Jamming
		Flooding
		Buffer overflow

vulnerability of the MAVLink protocol due to non-encrypted messages.

B. PROPOSED ATTACK SCENARIO

Here, by exploiting the vulnerability of the MAVLink protocol, we propose a methodology to disable a UAV. In particular, we exploit the fact that the MAVLink message is not encrypted. Accordingly, after sniffing the UAV network packets, we inject packets to disable the UAV. We consider a UAV system in which the UAV and GCS are connected via a network an the attacker is already hacked into the network, which is possible by various existing methods.

The attack scenario is as follows: In order to decide an attack target, it is necessary to have information on the hosts connected to the network. Therefore, an attacker operates in the promiscuous mode and obtain all the packets and sets the target. The attacker obtains the GCS and UAV packets by using an ARP poisoning attack, which sends fake ARP information to the host and makes the packet to be forwarded to the attacker.

By executing packet sniffing on the drone network, an attack can get the MAVLink packets. There are 160 kinds of common MAVLink packets; these packets send UAV state information or GCS commands in the MAVLink payload. By analyzing the packets to be transmitted, it is possible to identify whether the UAV is currently in flight, the state of the battery, what mission is being executed.

Based on the information, the attacker can identify the actual state of the UAV and can disable the UAV by sending malicious packets to the UAV. In this study, we use ICMP flooding attack as well as packet injection attack which exploits the vulnerability of the MAVLink waypoint protocol. Fig. 5 summarizes the overall procedure of the attack scenarios.

C. VULNERABILITY OF MAVLINK PROTOCOL TO FLOODING ATTACK

Internet control message protocol (ICMP) checks the connection status of the hosts in the network and reports when there is a problem with packet transfer. Using the ping command with Windows command or Linux kernel, an ICMP message can be sent. When sending an ICMP message, the sender will send an ICMP request packet to the receiver. Then, the receiver that has successfully received the request message will respond to the sender. If the sender sends a large number of request messages, the receiver will be overloaded to check and send replies. In this way, the ICMP flooding attack overloads the target system and invalidates the service. In section IV, we verify the effect of an ICMP flooding attack on a UAV by conducting ICMP flooding attack.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2863237, IEEE Access

D. VULNERABILITY OF MAVLINK WAYPOINT PROTOCOL TO PACKET INJECTION ATTACK

When using GCS to control the UAV, UAV executes the mission commands sent by GCS. At this time, mission commands are executed based on the waypoint protocol [30] in the MAVLink protocol. Fig. 6 shows the MAVLink waypoint protocol procedure. When the user completes the mission commands setting, the GCS sends information on the total number of missions as a MISSION_COUNT (N) message. Upon receiving this message, the UAV requests the first mission information using the MISSION_REQUEST (0) message. In response to this message, the GCS sends the first mission information with a MISSION_ITEM (0) message. In this way, the GCS sends a total of N pieces of mission information to the UAV. Upon completion of the mission information transfer, the UAV transmits a MISSION_ACK message to the GCS to notify that the transmission is completed.

We exploit the vulnerability of the waypoint protocol and carry out experiments with packet injection attack. When the GCS sends a MISSION_COUNT (N) packet, the UAV erases the stored mission information and prepares to receive new mission commands. Using these features, we conduct the experiment scenario as follows. Because the attacker had intruded into the network, the attacker is able to eavesdrop the information between GCS-UAV and obtain the mission information. After this, when the UAV executes the mission and starts the flight, the attacker sends an eavesdropped MISSION_COUNT (N) packet to the UAV and initialize the mission information. UAV sends MISSION_REQUEST to GCS to request mission information, but GCS has already sent mission information so it will not transmit. Therefore, the UAV enters a standby state waiting for mission information. In section IV, we empirically verify that the UAV under packet injection attack is disabled.

E. PACKET MONITORING AND INJECTION

In order to decide on an attack target, it is necessary to have information about the hosts connected to the network. Using Cain & Abel [24] as a network sniffing tool operating on Windows OS, we can obtain information on the hosts connected to the network. We usd Cain & Abel to learn the network IP address of the UAV and the GCS. Also, we obtain the GCS and UAV packets by using an ARP poisoning attack, which sends fake ARP information to the host and causes the packet to be forwarded to the attacker. Therefore, in UAV networks, packets of UAV and GCS can be transmitted to an attacker.

Jpcap [25] is a Java-based library that captures network packets. Using Jpcap to monitor the state of the UAV, in this study we develop a packet capture tool. Fig. 7 shows the developed program. As shown in Fig. 7, the program shows the network interface, source IP address, destination IP address and payload. The payload indicates the type of MAVLink data, which makes it possible to check the Message_ID of the MAVLink data. Using this program, we can estimate the state of the UAV in real time. For example, it is possible to confirm the MISSION_SET_CURRENT packet and determine what mission is currently being executed and whether or not the UAV is in flight. Therefore, we can know when to attack the UAV by monitoring the state information of UAV.

We use Packet Sender [26] to inject attack packets into the UAV. This program can send UDP and TCP network packets. Using this program, it is possible to transfer packets by changing to the payload desired by the user.

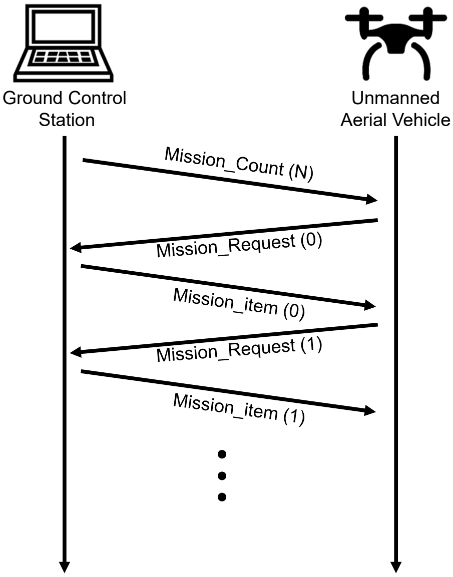
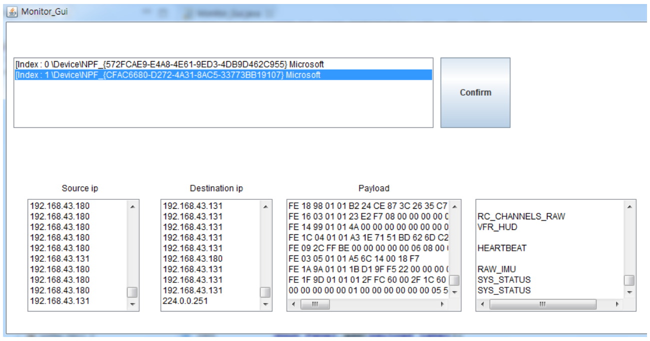


FIGURE 6. MAVLink waypoint protocol procedure.



[Index : 0 \Device\NPF_{572FCAE9-E4A8-4E61-9ED3-4DB9D462C955} Microsoft
[Index : 1 \Device\NPF_{CFAC6680-D272-4A31-8AC5-33773BB19107} Microsoft

Source ip	Destination ip	Payload	
192.168.43.180	192.168.43.131	FE 18 98 01 01 B2 24 CE 87 3C 26 35 C7	
192.168.43.180	192.168.43.131	FE 16 03 01 01 23 E2 F7 08 00 00 00 00 0	RC_CHANNELS_RAW
192.168.43.180	192.168.43.131	FE 14 99 01 01 4A 00 00 00 00 00 00 00 0	VFR_HUD
192.168.43.180	192.168.43.131	FE 1C 04 01 01 A3 1E 71 51 BD 62 6D C2	
192.168.43.180	192.168.43.131	FE 09 2C FF BE 00 00 00 00 00 06 08 00	HEARTBEAT
192.168.43.131	192.168.43.180	FE 03 05 01 01 A5 6C 14 00 18 F7	
192.168.43.180	192.168.43.131	FE 1A 9A 01 01 1B D1 9F F5 22 00 00 00	RAW_IMU
192.168.43.180	192.168.43.131	FE 1F 9D 01 01 01 2F FC 60 00 2F 1C 60	SYS_STATUS
192.168.43.180	192.168.43.131	00 00 00 00 00 01 00 00 00 00 00 05 5	SYS_STATUS
192.168.43.131	224.0.0.251		

FIGURE 7. Monitoring program developed using Jpcap library.

IV. ATTACK IMPLEMENTATION

A. TESTBED CONFIGURATION

In order to perform experiments in the UAV network, we construct a testbed as shown in Fig. 8. We install hostapd [27] in raspberry-pi3 for the wireless access point, which will be used for connecting the UAV and GCS. We use 3DR X8 + drone in Fig. 9 for our experiments. Since this drone uses pixhawk, it can be controlled using the MAVLink protocol. In order to allow the drone to connect to the access point, we use raspberry-pi3, which includes installing mavproxy [28].

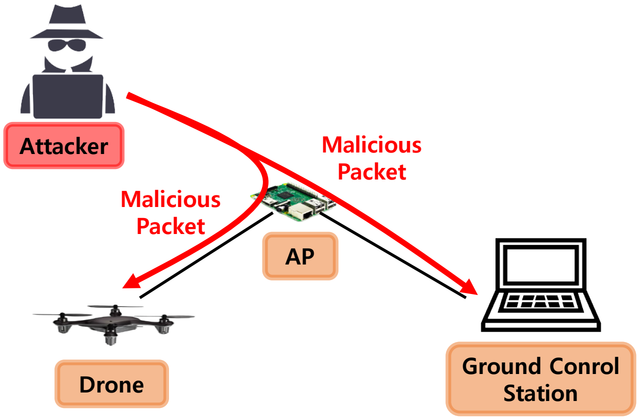


FIGURE 8. Testbed configuration with AP, GCS, and drone.

FIGURE 9. 3DR X8+ drone used for experiments.

The GCS used for the experiment is the mission planner [29] as shown in Fig. 10.

B. ICMP FLOODING ATTACK

In an environment connected to an access point, we carry out experiments with the effect of an ICMP flooding attack on a UAV. First, when the attacker sends ICMP request packets to the GCS and the UAV at 7 Mb/s. Fig. 11(a) shows the change in the inter-reception time of sensor values when sending ICMP packets to the UAV. In this experiment, we select pitch values for the UAV. The normal case is shown in Fig. 11(a); it can be easily confirmed that the inter-reception time does not greatly deviate from the average of 0.24, but that this value greatly changes in the case of ICMP attack. In the normal case, the variance of the inter-reception time is measured at about 0.238×10^{-3}; in the case of ICMP attack, the variance of the inter-reception time is measured at about 8.4×10^{-3}. The variance of the inter-reception time during the ICMP attack is about 35 times larger than that of the normal case.

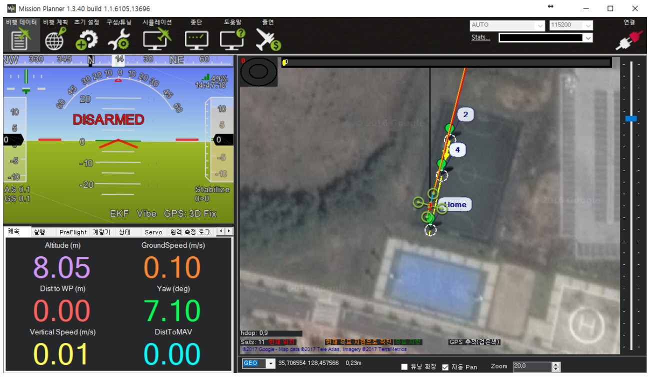


FIGURE 10. Mission planner used for experiments.

Fig. 11(b) shows the change in the inter-reception time of pitch values when sending ICMP packets to the GCS. In this figure, the variance of the inter-reception time in the normal case is measured at about 0.238×10^{-3}; in the case of ICMP attack, the variance of the inter-reception time is measured about 2.42×10^{-3}. The variance of the inter-reception time for the ICMP attack is about 10 times larger than that of the normal case. In this experiment, we can confirm that the variance of the packet inter-reception time is larger for an ICMP flooding attack on the UAV.

We also conduct an experimental ICMP flooding attack on a UAV that was executing a mission. In this experiment, we confirm that the UAV's sensor values are not transmitted well, and the mission commands delivered by the GCS are also not transferred properly. A heartbeat message is sent between the GCS and the UAV in one second period to maintain the connection. If the heartbeat message is not received for longer than 3 seconds, the UAV will operate in failsafe mode. In this experiment, because of the ICMP flooding attack, the UAV can not receive a heartbeat message within 3 seconds.

C. PACKET INJECTION ATTACK

We carry out experiments to transmit MISSION_COUNT (N) packets to the UAV executing its mission. As a result of the experiment, we can confirm that the UAV starts to hover immediately after receiving the MISSION_COUNT (N) packet. This is because all of the mission information that the GCS has sent before are deleted due to the MISSION_COUNT (N) packet that has been forwarded.

Fig. 13 shows the console screen of the UAV mavproxy that receives the packet of MISSION_COUNT (N). In Fig. 13, "not loading waypoint" appears on the console screen after receiving the MISSION_COUNT (N) packet while waypoint 2 is executing. In this state, the UAV continuously hovers unless the battery is exhausted or a new mission command is transmitted. When the UAV is in the hovering state, if an attacker injects a packet containing mission information, the UAV will execute the mission sent by the attacker. Our experiment can be found in [32]. Other UAV attacks usually cause unpredictable secondary damage due to the UAV's ground crash, while our attack does not cause crashes because the UAV is forced to be hovering around.

Fig. 12 shows how the ground speed is different without attack and with attack. The ground speed is the relative speed of UAVs with respcet to the ground. Thus, the ground speed is an effective indicator to show the behavior of UAVs, i.e.,

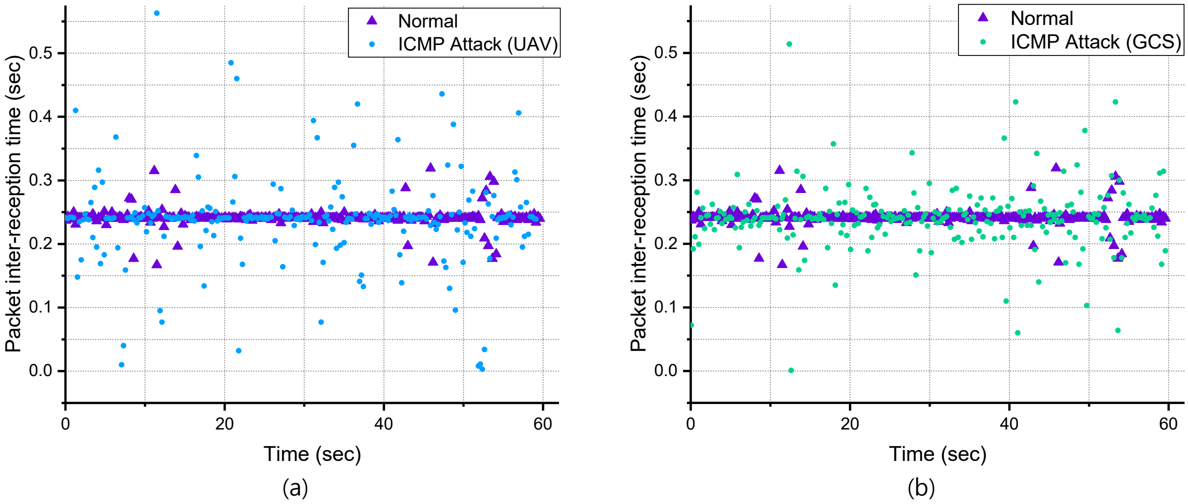


FIGURE 11. Packet inter-reception time under normal operation and under ICMP attack on UAV and GCS.

whether it is hovering around or carrying out its mission. Fig. 12(a) shows the ground speed of the UAV without any attack. When the time instant is around 35 second in Fig. 12(a), the ground speed decreases due to waypoint change of the UAV mission. Fig. 12(b) shows the ground speed of the UAV under a packet injection attack. We perform the packet injection attack just before the waypoint of the UAV is changed. In Fig. 12(b), we can see that the UAV stops the mission under the packet injection attack and hovers around in a few seconds.

D. SOFTWARE IN THE LOOP (SITL) SIMULATOR

Here, with the software in the loop (SITL) simulator [31], the experiment scenario conducted in Section IV.B and C are performed in the same way. We use the mission planner as the GCS and connected the UAV to mavproxy in SITL.

First, we conduct experiments with SITL on how ICMP flooding affects the UAV. As in the previous experiment, it is confirmed that the packet inter-reception time greatly fluctuates.

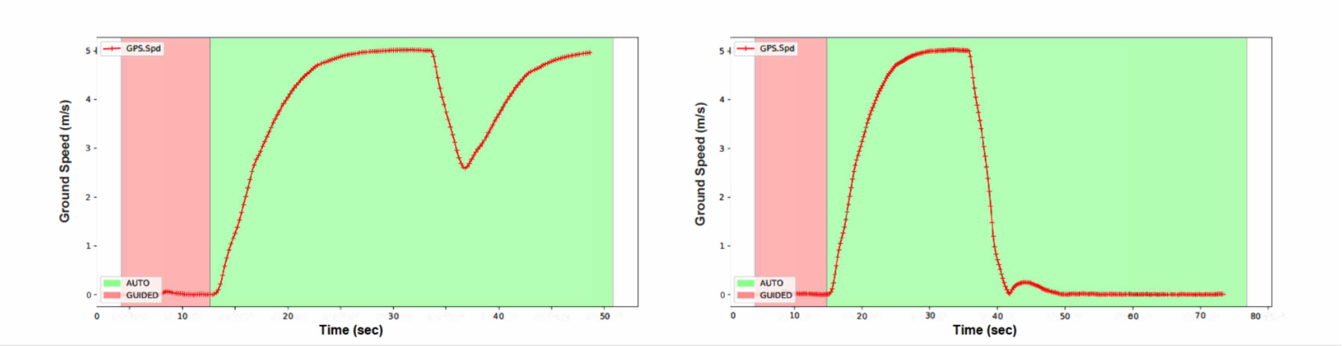
In addition, the same scenario as used for the packet injection experiment conducted previously is used with SITL. Fig. 14 shows the packet injection experiment in SITL. Fig. 15 shows the UAV mavproxy console screen after execution of SITL. As in the previous experiment, when the UAV receives the MISSION_COUNT (N) packet, we can confirm that “not loading waypoints” is displayed on the command screen. Similarly, our experiment with SITL can be found in [33].

V. RELATED WORK

One way to disable a UAV is to use a sensor and hardware attack on the UAV, or a network attack. Sensor and hardware attacks make use of UAV sensor vulnerabilities to disable the UAV. In general, communication link jamming and GPS spoofing are used for sensor attacks in UAV systems. Jamming prevents the communication link between the UAV and the GCS or the controller from correct operation so that the control message of the UAV cannot be transmitted. In the structure of the UAV system shown in Fig. 3, GPS spoofing is a scheme utilizing the vulnerability of the communication between the GPS satellite and the UAV GPS sensor. A GPS spoofing attack is used to trick the UAV by broadcasting a fake GPS signal [9, 15]. In the case of a real GPS signal, the distance between the satellite and the sensor is long, so the GPS signal power can be weakened. Thus, it is possible to transmit fake GPS information to the UAV by generating GPS signals near the UAV. In [18], the authors study a GPS spoofing attack for the GPS receiver. These attacks either require special equipment or has a limited attack range, while our attack method can be made without any special equipment and distance constraints.

In [10], the authors conduct research to disable a UAV by attacking access point in Wi-Fi networks. In this research, the authors use the vulnerability of wired equivalent privacy (WEP), which is one of the WiFi security protocols. WEP encryption has a vulnerability that makes it possible to crack the pre-shared key by collecting a certain amount of data. In particular, using the password crack tool aircrack-ng, it is easy to crack the pre-shared key value in WEP encryption. Using aircrack-ng, the authors disable the UAV by sending de-authentication packets to the UAV. This attack is only applied to UAVs that use Wi-Fi as a communication protocol, while our attack method can be applied to any UAV systems using the MAVLink protocol.

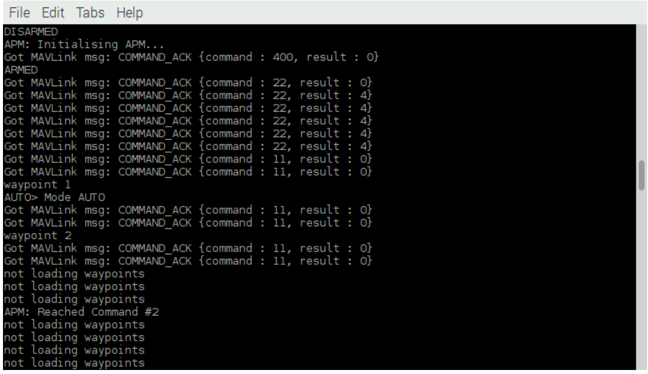
In [19], the authors carry out an experiment to disable a UAV using a man-in-the-middle attack. In this system, the authors use the Zigbee API mode, which can send broadcast packets to UAV networks. The broadcast packets collect the initial vector values, which are used to crack the WEP.



(a)

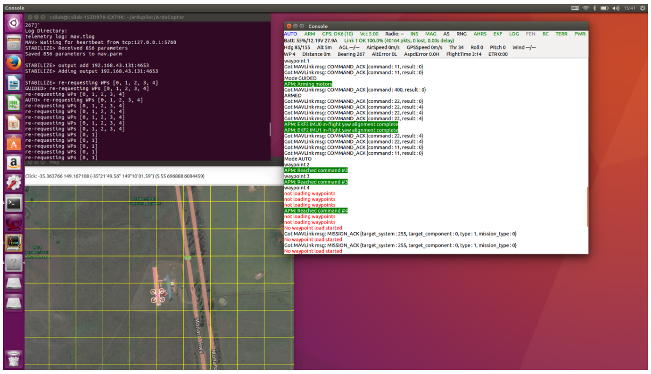
(b)

FIGURE 12. Ground speed under normal operation and under packet injection attack.



```
DISARMED
APM: Initialising APM...
Got MAVLink msg: COMMAND_ACK {command : 400, result : 0}
ARMED
Got MAVLink msg: COMMAND_ACK {command : 22, result : 0}
Got MAVLink msg: COMMAND_ACK {command : 22, result : 4}
Got MAVLink msg: COMMAND_ACK {command : 22, result : 4}
Got MAVLink msg: COMMAND_ACK {command : 22, result : 4}
Got MAVLink msg: COMMAND_ACK {command : 22, result : 4}
Got MAVLink msg: COMMAND_ACK {command : 22, result : 4}
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
waypoint 1
AUTO> Mode AUTO
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
waypoint 2
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
not loading waypoints
not loading waypoints
not loading waypoints
APM: Reached Command #2
not loading waypoints
not loading waypoints
not loading waypoints
not loading waypoints
```

FIGURE 13. Mavproxy command screen.

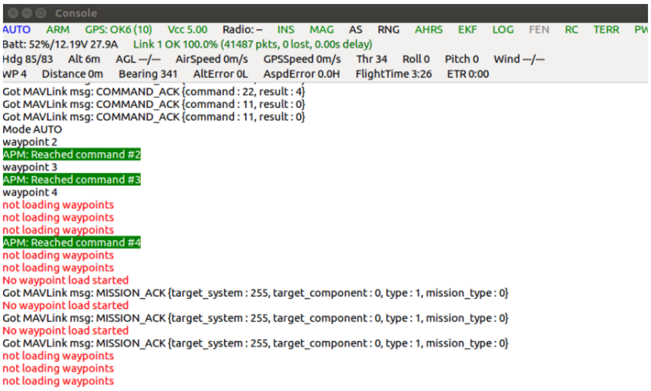

Console
csilab@csilab-15ZD970-GX70K: ~/ardupilot/ArduCopter
Map
Jerrabomberra Grassland Nature R
West Jerrabomberra Nature Reserve

```
267]'
Log Directory:
Telemetry log: mav.tlog
MAV> Waiting for heartbeat from tcp:127.0.0.1:5760
STABILIZE> Received 856 parameters
Saved 856 parameters to mav.parm

STABILIZE> output add 192.168.43.131:4653
STABILIZE> Adding output 192.168.43.131:4653

STABILIZE> re-requesting WPs [0, 1, 2, 3, 4]
GUIDED> re-requesting WPs [0, 1, 2, 3, 4]
re-requesting WPs [0, 1, 2, 3, 4]
AUTO> re-requesting WPs [0, 1, 2, 3, 4]
re-requesting WPs [0, 1, 2, 3, 4]
re-requesting WPs [0, 1, 2, 3, 4]
re-requesting WPs [0, 1, 2, 3, 4]
re-requesting WPs [0, 1, 2, 3, 4]
re-requesting WPs [0, 1, 2, 3, 4]
re-requesting WPs [0, 1]
re-requesting WPs [0, 1]
re-requesting WPs [0, 1]
re-requesting WPs [0, 1]
re-requesting WPs [0, 1]
```

```
Click: -35.363766 149.167108 (-35°21'49.56" 149°10'01.59") (S 55 696888 6084459)
```

```
AUTO   ARM   GPS: OK6 (10)   Vcc 5.00   Radio: --   INS   MAG   AS   RNG   AHRS   EKF   LOG   FEN   RC   TERR   PWR
Batt: 55%/12.19V 27.9A   Link 1 OK 100.0% (40164 pkts, 0 lost, 0.00s delay)
Hdg 85/155   Alt 5m   AGL --/--   AirSpeed 0m/s   GPSSpeed 0m/s   Thr 34   Roll 0   Pitch 0   Wind --/--
WP 4   Distance 0m   Bearing 267   AltError 0L   AspdError 0.0H   FlightTime 3:14   ETR 0:00
waypoint 1
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
Mode GUIDED
APM: Arming motors
Got MAVLink msg: COMMAND_ACK {command : 400, result : 0}
ARMED
Got MAVLink msg: COMMAND_ACK {command : 22, result : 0}
Got MAVLink msg: COMMAND_ACK {command : 22, result : 4}
Got MAVLink msg: COMMAND_ACK {command : 22, result : 4}
Got MAVLink msg: COMMAND_ACK {command : 22, result : 4}
APM: EKF2 IMU0 in-flight yaw alignment complete
APM: EKF2 IMU1 in-flight yaw alignment complete
Got MAVLink msg: COMMAND_ACK {command : 22, result : 4}
Got MAVLink msg: COMMAND_ACK {command : 22, result : 4}
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
Mode AUTO
waypoint 2
APM: Reached command #2
waypoint 3
APM: Reached command #3
waypoint 4
not loading waypoints
not loading waypoints
not loading waypoints
APM: Reached command #4
not loading waypoints
not loading waypoints
No waypoint load started
Got MAVLink msg: MISSION_ACK {target_system : 255, target_component : 0, type : 1, mission_type : 0}
No waypoint load started
Got MAVLink msg: MISSION_ACK {target_system : 255, target_component : 0, type : 1, mission_type : 0}
No waypoint load started
```

FIGURE 14. Experiment using SITL simulator.



```
AUTO   ARM   GPS: OK6 (10)   Vcc 5.00   Radio: --   INS   MAG   AS   RNG   AHRS   EKF   LOG   FEN   RC   TERR   PW
Batt: 52%/12.19V 27.9A   Link 1 OK 100.0% (41487 pkts, 0 lost, 0.00s delay)
Hdg 85/83   Alt 6m   AGL --/--   AirSpeed 0m/s   GPSSpeed 0m/s   Thr 34   Roll 0   Pitch 0   Wind --/--
WP 4   Distance 0m   Bearing 341   AltError 0L   AspdError 0.0H   FlightTime 3:26   ETR 0:00
Got MAVLink msg: COMMAND_ACK {command : 22, result : 4}
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
Mode AUTO
waypoint 2
APM: Reached command #2
waypoint 3
APM: Reached command #3
waypoint 4
not loading waypoints
not loading waypoints
not loading waypoints
APM: Reached command #4
not loading waypoints
not loading waypoints
No waypoint load started
Got MAVLink msg: MISSION_ACK {target_system : 255, target_component : 0, type : 1, mission_type : 0}
No waypoint load started
Got MAVLink msg: MISSION_ACK {target_system : 255, target_component : 0, type : 1, mission_type : 0}
No waypoint load started
Got MAVLink msg: MISSION_ACK {target_system : 255, target_component : 0, type : 1, mission_type : 0}
not loading waypoints
not loading waypoints
not loading waypoints
```

FIGURE 15. UAV mavproxy console screen executed in SITL simulator.

As in [10], the authors used the vulnerability of WEP to hack the UAV. This attack method can only attack a specific manufacturer's UAV. On the other hand, since the MAVLink protocol is a de factor standard, our attack can be considered a more general approach.

In [20], a method to hijack a UAV using the vulnerability of the MAVLink protocol is proposed. When using the telemetry module to control the UAV via MAVLink, it is necessary to enter the NetID to connect to the UAV. Therefore, if the NetID is known, it is easy to hijack the UAV. Using this, the authors of [21] execute an attack by using an antenna with the same NetID to repeatedly send malicious MAVLink packets. Unlike this approach, our attack method does not require any additional information such as NetID.

In [22, 23], the authors hijack a UAV using the vulnerability of the AR drone. In particular, in [22], the authors use port scanning of the FTP port, and then sent a malicious code to the UAV to access the UAV's private pictures and information without permission. Also, in [23], the authors perform an attack using an AR drone's telnet port vulnerability to re-install the shell script and restart the AR drone. In this way, they easily stole the authority of the AR drone.

VI. CONCLUSIONS

In this paper, we have empirically studied the vulnerability of the MAVLink protocol. By exploiting the unencrypted messages of the MAVLink protocol, we have devised an attack methodology to disable a UAV. In our experiments, first we have studied ICMP flooding scenario, in which we can confirm that the packet inter-reception time significantly fluctuates which can be fatal to the UAV. We have further carried out packet injection experiments, in which we have exploited the vulnerability of the waypoint protocol to send malicious packets for deleting mission information of the UAV. Consequently, under the packet injection attack, the UAV on mission has stopped and hovered because of deleted mission information. In summary, we have found out the vulnerability of the MAVLink protocol and have verified them with empirical study.

REFERENCES

[1] M. Gharibi, R. Boutaba, and S.L. Waslander, "Internet of drones," IEEE Access, vol. 4, pp. 1148-1162, Mar. 2016.

[2] S. Waharte and N. Trigoni, "Supporting search and rescue operations with UAVs," in Proc. Int. Conf. Emerging Security Technology, Canterbury, U.K., 2010, pp. 142–147.

[3] S. M. Adams and C. J. Friedland, "A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management," presented at the 9th Int. Workshop on Remote Sensing for Disaster Response, California, USA, Sept. 14-16, 2011.

[4] A. J. S. McGonigle et al., "Unmanned aerial vehicle measurements of volcanic carbon dioxide fluxes," Geophysical research letters, to be published. DOI: 10.1029/2007GL032508.

[5] Amazon prime-air projects. Accessed on: Jan. 14, 2018. [Online]. Available: https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011

[6] Fleetlights. Accessed on: Jan. 14, 2018. [Online]. Available: https://www.directline.com/fleetlights

[7] Matternet. Accessed on: Jan. 14, 2018. [Online]. Available: https://mttr.net

[8] MALink protocol. Accessed on: Jan. 14, 2018. [Online]. Available: http://qgroundcontrol.org/mavlink/start

[9] K. Domin, E. Marin, and I. Symeonidis, "Security Analysis of the Drone Communication Protocol: Fuzzing the MAVLink protocol," in Proc. Symposium on Information Theory in the Benelux, Louvain-la-Neuve, Belgium, 2016, pp. 198–204.

[10] C. Rani, H. Modares, R. Sriram, D. Mikulski, and F. L. Lewis, "Security of unmanned aerial vehicle systems against cyber-physical attacks," The Journal of Defense Modeling and Simulation., vol. 13, issue. 3, pp. 331-342, July. 2016.

[11] O. Alberto and M. Valleri, "Man in the middle attacks," in Proc. Blackhat Conf. Europe, 2013, [Online]. Available: http://blackhat.com/presentations/bh-europe-03/bh-europe-03-valleri.pdf

[12] J. A. Marty, "Vulnerability analysis of the mavlink protocol for command and control of unmanned aircraft," M.S. thesis, Dept. Electrical Computer Eng., Air Force Institute of Technology, Ohio, United States, 2013.

[13] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring Internet denial-of-service activity," ACM Trans. on Computer Systems., vol. 24, issue. 2, pp. 115–139, May. 2006.

[14] M. D. Nguyen, N. Dong, and A. Roychoudhury, "Security analysis of unmanned aircraft systems," National Univ. of Singapore, Singapore, Tech. Rep. TRA1/17, Jan. 2017.

[15] K. Hartmann and C. Steup, "The vulnerability of UAVs to cyber attacks-An approach to the risk assessment," in Proc. Cyber Conflict, Tallinn, Estonia, 2013 pp. 1–23.

[16] A. Y. Javaid, W. Sun, V. K. Devabhaktuni, and M. Alam, "Cyber security threat analysis and modeling of an unmanned aerial vehicle system," in Proc. IEEE Conf. on Technol. for Homeland Security, Waltham, MA, USA, 2012 pp. 585–590.

[17] K. M. Mansfield, T. J. Eveleigh, T. H. Holzer, and S. Sarkani, "DoD comprehensive military unmanned aerial vehicle smart device ground control station threat model," Defense Acquisition Research Journal, vol. 22, no. 2, pp. 240–273, Apr. 2015.

[18] N. O. Tippenhauer, C. Popper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful GPS spoofing attacks," in Proc. ACM conf. on Computer and commun. security, New York, USA, 2011 pp. 75–86.

[19] N. M. Rodday, R. D. O. Schmidt, and A. Pras, "Exploring security vulnerabilities of unmanned aerial vehicles," in Proc. IEEE/IFIP Network Operations and Management Symposium, Istanbul, Turkey, 2016 pp. 993–994.

[20] Hijacking drones with a MAVLink exploit. Accessed on: Jan. 14, 2018. [Online]. Available: http://diydrones.com/profiles/blogs/hijacking-quadcopters-with-a-mavlink-exploit

[21] K. Highnam, K. Angstadt, K. Leach, W. Weimer, A. Paulos, and P. Hurley, "An uncrewed aerial vehicle attack scenario and trustworthy repair architecture," in Proc. IEEE/IFIP Int. Conf. on Dependable Systems and Networks Workshop, Toulouse, France, 2016 pp. 222–225.

[22] F. Samland, J. Fruth, M. Hildebrandt, T. Hoppe, and J. Dittmann, "AR. drone: Security threat analysis and exemplary attack to track persons," in Proc. IS&T/SPIE Electronic Imaging, California, United States, 2012 pp. 83010G–83010G.

[23] J. S. Pleban, R. Band, and R. Creutzburg, "Hacking and securing the AR. Drone 2.0 quadcopter: investigations for improving the security of a toy," in Proc. IS&T/SPIE Electronic Imaging, California, United States, 2014 pp. 90300L–90300L.

[24] Cain & abel. Accessed on: Jan. 14, 2018. [Online]. Available: http://www.oxid.it/cain.html

[25] Jpacap. Accessed on: Jan. 14, 2018. [Online]. Available: http://jpcap.gitspot.com/index.html

[26] Packet sender. Accessed on: Jan. 14, 2018. [Online]. Available: https://packetsender.com

[27] Hostapd. Accessed on: Jan. 14, 2018. [Online]. Available: https://w1.fi/hostapd

[28] Mavproxy. Accessed on: Jan. 14, 2018. [Online]. Available: http://ardupilot.github.io/MAVProxy/html/index.html

[29] Mission planner. Accessed on: Jan. 14, 2018. [Online]. Available: http://ardupilot.org/planner

[30] MAVLink waypoint protocol. Accessed on: Jan. 14, 2018. [Online]. Available: http://qgroundcontrol.org/mavlink/waypoint_protocol

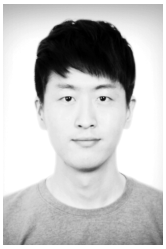
[31] Software in the loop simulator. Accessed Jan. 14, 2018. [Online]. Available: http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html

[32] Packet injection attack experiment. Accessed on: Jan. 30, 2018. [Online]. Available: https://youtu.be/BA7NicJg4os

[33] Packet injection attack experiment with SITL simulator. Accessed on: Jan. 30, 2018. [Online]. Available: https://youtu.be/o7yrj7XqOgw

YOUNG-MIN KWON received the B.S. degree in information and communication engineering from Korea University of Technology and Education, South Korea, in 2016, and the M.S. degree in information and communication engineering from Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, South Korea, in 2018. His research interests include resilient cyber-physical system.

JAEMIN YU received the B.S. degree in Computer System from Samyook University, South Korea, in 2017. He is currently pursuing the M.S. degree with the Cyber-Physical Systems Integration Laboratory, Daegu Gyeongbuk Institute of Science and Technology (DGIST). His research interests include resilient cyber-physical system.



BYEONG-MOON CHO received the B.S. degree in information and communication engineering from Chungbuk University, South Korea, in 2013. He is currently pursuing the Ph.D. degree with the Cyber-Physical Systems Integration Laboratory, Daegu Gyeongbuk Institute of Science and Technology (DGIST). His research interests include resilient cyber-physical system, congestion control in vehicular ad-hoc networks and Coexistence of heterogeneous networks.

YONGSOON EUN (M'03) received the B.A. degree in mathematics, and the B.S. and M.S.E. degrees in control and instrumentation engineering from Seoul National University, Seoul, Korea, in 1992, 1994, and 1997, respectively, and the Ph.D. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, MI, USA, in 2003. From 2003 to 2012, he was a Research Scientist with the Xerox Innovation Group, Webster, NY, USA, where he worked on a number of subsystem technologies in the xerographic marking process and image registration method in production inkjet printers. Since 2012, he is an Associate Professor with the Department of Information and Communication Engineering, DGIST, Korea. His research interests include control systems with nonlinear sensors and actuators, geometric control of quadrotors, communication network, and resilient cyber-physical systems.

KYUNG-JOON PARK (M'05) received the B.S. and M.S. degrees in electrical engineering from the School of Electrical Engineering, and the Ph.D. degree in electrical engineering and computer science from Seoul National University, Seoul, Korea, in 1998, 2000, and 2005, respectively. From 2005 to 2006, he was a Senior Engineer with the Samsung Electronics, Suwon, Korea. From 2006 to 2010, he was a Postdoctoral Research Associate with the Department of Computer Science, University of Illinois at Urbana-Champaign (UIUC), Champaign, IL, USA. He is currently an Associate Professor with the Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, Korea. His research interests include resilient cyber-physical systems and smart factory.