Master's Thesis
석사 학위논문

An Empirical Study on the Attack and Defense of Unmanned Vehicle

Jiyoung Yoon(윤 지 영 尹 持 榮)

Department of
Information and Communication Engineering

DGIST

2020

An Empirical Study on the Attack and Defense of Unmanned Vehicle

Advisor: Professor Kyung-Joon Park
Co-advisor: Professor Kyoung-Dae Kim

by

Jiyoung Yoon
Department of Information and Communication Engineering
DGIST

A thesis submitted to the faculty of DGIST in partial fulfillment of the requirements for the degree of Master of Science in the Department of Information and Communication Engineering. The study was conducted in accordance with Code of Research Ethics[1]

05. 08. 2020

Approved by

Professor Kyung-Joon Park (signature)
(Advisor)

Professor Kyoung-Dae Kim (signature)
(Co-Advisor)

[1] Declaration of Ethical Conduct in Research**:** I, as a graduate student of DGIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

An Empirical Study on the Attack and Defense of Unmanned Vehicle

Jiyoung Yoon

Accepted in partial fulfillment of the requirements for the degree of Master of Science.

05. 08. 2020

Head of Committee Prof. Kyung-Joon Park (signature)

Committee Member Prof. Kyoung-Dae Kim (signature)

Committee Member Prof. Jihwan Choi (signature)

MS/IC
201822021

윤 지 영. Jiyoung Yoon. An Empirical Study on the Attack and Defense of Unmanned Vehicle. Department of Information and Communication Engineering. 2020. 33p. Advisors Prof. Kyung-Joon Park, Co-Advisors Prof. Kyoung-Dae Kim

ABSTRACT

One of the main applications of the Cyber-Physical System, the Unmanned vehicle is gradually expanding its use. Unmanned Aerial Vehicle (UAV), among unmanned vehicle, is used not only for cameras, emergency, and military purposes, but its negative effects are increasing also as its use expands. A terrorist outrage using UAVs in Saudi Arabia in the fall of 2019 is a well-known example. Therefore, research on disabling UAV is also becoming important. The UAV neutralization study can be divided into three phases. First, it is the identification of friend or foe stage that distinguishes whether UAVs are friendly or enemy. However, this step can be omitted in No-drone Zones, such as places where people are concentrated, places where major confidential facilities such as nuclear facilities are located, and places of privacy. The second step is to neutralize the UAV's actual mission. At this stage, the UAV is disabled mainly through network attacks such as jamming attacks and packet injection attacks, or through physical attacks such as nets. The third is a post-processing step to lead the UAV to safe area, that is, to prevent the UAV from flying again and to protect the surroundings from it. Previous UAV neutralization studies have focused on disabling UAV without considering the third phase. In this paper, we focused on the third stage, the post-processing stage, so that UAV can be neutralized. Robot Operating System is useful and used widely in UAV system, but there are also vulnerabilities. Therefore, disabling UAVs using this point and defense techniques are discussed in this paper.

Keywords: Unmanned Aerial Vehicle, Network Attack, Cyber-Physical System, Security

List of Contents

List of Tables

List of Figures

List of algorithms

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs), called drones, are increasingly being used in the construction, mining industry, life-saving activities, leisure, media, entertainment, and military. According to the Ministry of Land, Infrastructure and Transport, the global drone market is expected to grow by 29% annually to reach about $ 82 billion in 2026. Drones have emerged as one of the important applications of Cyber-Physical Systems (CPS), the theme of the 4th industrial revolution [1, 2].

In September, UAVs were used as terrorist attacks, including drone attacks on refineries and oil production bases in Saudi Arabia, as well as the death of the South Yemen army through suicide bombers. In addition, UAVs are also important for information security on missions on military purposes. So, the vulnerability of UAV network has been proved through a lot of experiments. Therefore, various defense methods for the vulnerabilities have been researched also [3-7].

CPS is an integrated system that observes and controls the three elements of a physical system, communication and computation, and the state of the physical system through a network. It is mainly used for unmanned vehicles, smart factories, medical devices, etc., and is expanding its scope of use. Robot Operating System (ROS) is also a system that integrates and manages applications, sensors, and robots into ROS through a network, helping robots with complex computation, hardware abstraction, and data management. ROS is also applied to robots such as unmanned vehicles and smart factories.

The UAV neutralization step can be divided into three main stages. The first step is to identify the UAV as an attack target as identification friend or foe step. However, this step can be omitted in NO-drone zones such as nuclear facilities, military bases, and densely populated areas. The second step is to actually disable the UAV. The unmanned aerial vehicle

is being studied through methods such as jamming, network attacks through protocol vulnerabilities, and spoofing [8-12]. The third step, a step after the neutralization of the UAV, is a post-processing step. While neutralizing the UAV, we conclude by eliminating secondary damage from UAV crash or hovering and recovering the UAV safely. If the UAV is neutralized in flight at high altitude, it causes serious secondary damage in case of human injury, property damage, or a fall in nuclear facilities. The first stage and the second stage are generally under study, but the third stage has not yet been studied well. In this paper, we will talk about the study of the third stage, including the second stage of the UAV.

II. BACKGROUND

2.1 Unmanned Aerial System (UAS)

Unmanned Aerial System (UAS) refers to all systems including unmanned aerial vehicles, communication equipment, ground stations, and communications. UAV can be controlled using various media such as RC transmitter, Bluetooth, Wi-Fi, etc., and UAV is connected to GCS (Ground Control Station) through the corresponding medium for flying. We configured UAS considering the network environment of UAV and GCS through Wi-Fi. GCS and UAV are connected to Wi-Fi and use MAVLink protocol. The MAVLink protocol is used to communicate not only between UAV and GCS, but also among various components mounted on the UAV, and is a representative protocol used by many UAVs as well as DJI and Parrot products. The MAVLink protocol is a header-oriented protocol and is a protocol suitable for UAVs with limited resources. The connected GCS can give UAV commands related to flight missions and monitor the UAV's flight status and status information of various sensors. Figure 2.1 shows the MAVLink protocol message. MAVLink is divided into MAVLink 1.0 and MAVLink 2.0. MAVLink 2.0 has many devices that are not yet supported, but it is backward compatible. Figure 2.2 shows QGroundControl as GCS in UAS configuration. Figure 2.3 shows the UAV used in this paper.

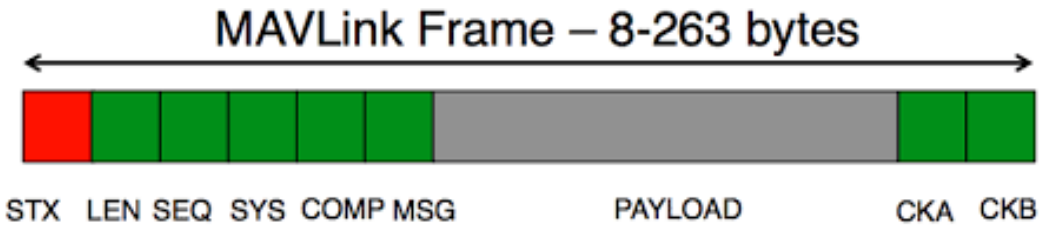


Figure 2. 1. MAVLink protocol message.

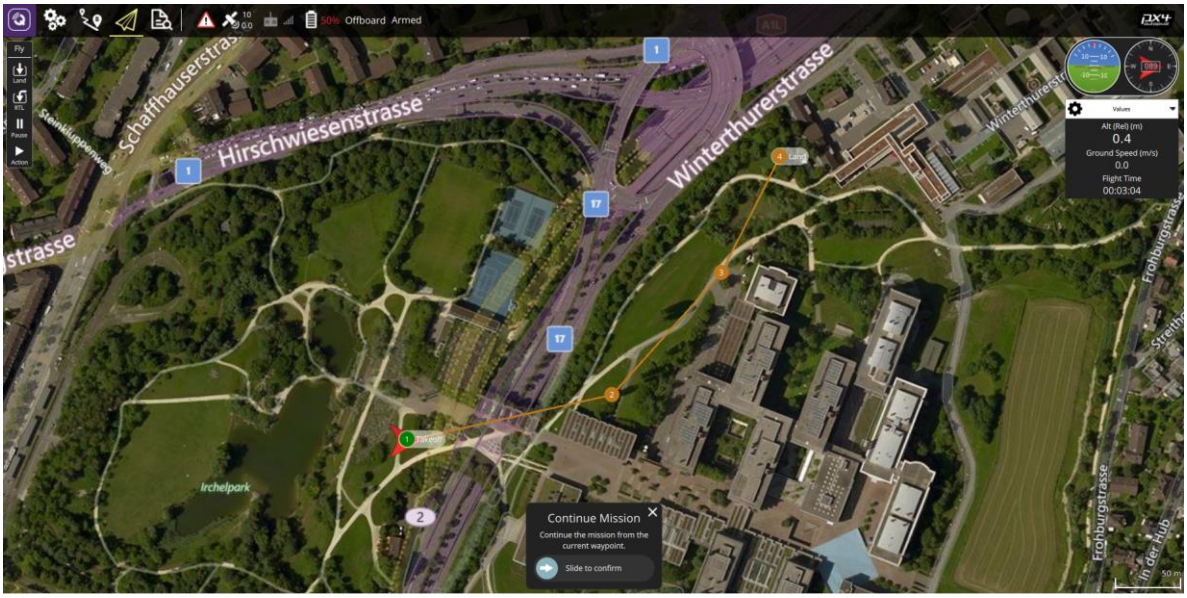


Figure 2. 2. QGroundControl as Ground Control Station (GCS).

Figure 2. 3. Unmanned Aerial Vehicle (UAV).

2.2 Robot Operating System (ROS)

Robot Operating System (ROS) is a robot open source meta operating system that implements functions mainly used in robots, including hardware abstraction, and provides message transfer and file management between processes. ROS structure can be briefly expressed as in Figure 2.4. ROS is largely composed of ROS master, publisher node, and subscriber node. The ROS master receives information related to the connection from the nodes that want to connect in order to initially connect the nodes. Nodes that exchange each one's IP address with the ROS master later connect directly without the ROS master. At this time, the node that produces and provides data is called the publisher node, and publishes the data on a predetermined topic. The Subscriber node can subscribe to data published by the publisher node from a desired topic.

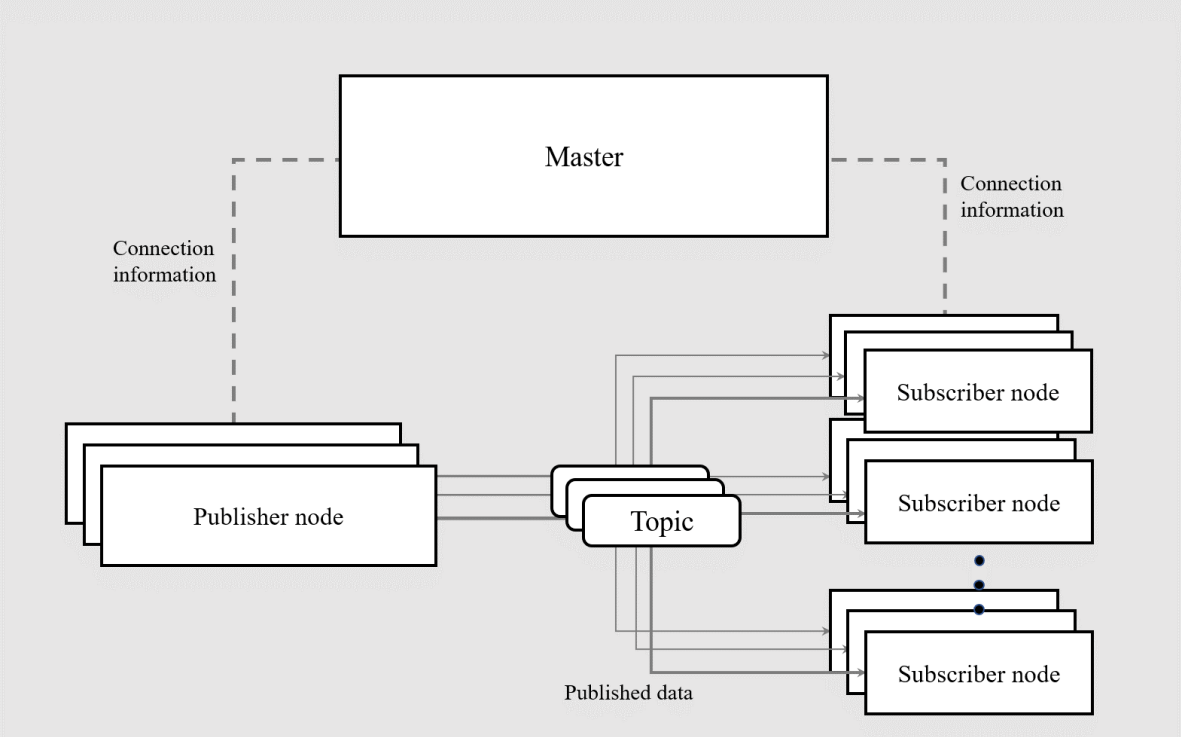


Figure 2. 4 Robot Operating System structure.

III. RELATED WORK

In UAS, UAV, a physical system equipped with sensors and actuators, is connected to the GCS using various media such as Wi-Fi to send and receive information about control and monitoring. It is assumed that UAV and GCS are connected in a Wi-Fi environment to transmit information using the MAVLink protocol.

However, the MAVLink protocol is a protocol for communication in unmanned vehicles with limited resources and is vulnerable to network attacks. There is a study that disables the UAV by using Mission_count packet, which is one of the messages of the MAVLink protocol [3]. In addition, UAV and GCS were simultaneously attacked to neutralize UAV, and GCS was not aware of disabling UAV, thereby slowing the response in GCS as the control system. Also, in research [4], UAS is configured by using ESP8266 as an AP in UAV to use the MAVLink protocol. At this time, the UAV is disabled by using the vulnerability in MavESP8266, the firmware proposed by PX4 and Ardupilot. So, the process of verifying the identity of the GCS connected to the MavESP8266 was added to make up for the vulnerability.

In a machine in a smart factory, a system for autonomous driving of an unmanned mobile vehicle, and a smart medical device system, the ROS is a system commonly used to connect a network and a machine that is a hardware and an application for control. However, since there is no authentication or authentication system for security, vulnerability research has been continued [13].

In particular, when attacking using the vulnerabilities of ROS, in the aforementioned unmanned mobile vehicles or smart factories, it is important to defend against attacks because they can cause fatal risks even in a short period of time to attack. In study [13, 14], research was conducted to apply key exchange for authentication to ROS. In research [13,

15], in ROS, security is improved by adding a security element by adding a hand shake process in the connection process between the ROS master and each node, or by applying a protocol certified for security in other applications. During data exchange in ROS, data was transmitted through data encryption and information was secured [16]. In addition, there have been studies to improve security in ROS by modifying the architecture of ROS or by other methods [17]. In addition, by adding a monitoring node for security to the ROS, the system is continuously monitored to verify that unauthorized nodes are connected to the ROS application and there are also researches to improve security of ROS by adding security tools to ROS [19, 20].

IV. PROPOSED METHOD

4.1 Proposed attack method

Cyber-Physical System (CPS) generally refers to a system in which sensors and actuators are networked and interacted with. CPS is widely applied to smart factories, medical systems, and autonomous unmanned vehicles. However, there is a point that the network, which is the data transmission layer of CPS configuration, can be vulnerable to network attacks as it is exposed to the outside. UAV, which is one of the representative examples of CPS, is gradually expanding its use for leisure, photography, emergency aid, and military use. However, UAV also inherits the network vulnerability of CPS, and research is underway.

As the utilization of UAV is gradually widening, malicious use is also increasing. Therefore, the research to neutralize UAV is also becoming important. A representative UAV neutralization study disables UAV by disabling one or more of the functions of UAV through jamming. In addition, there is a study of disabling using the vulnerability of the MAVLink protocol mainly used by UAV. The MAVLink protocol is a protocol that has advantages in lightness of calculation and throughput according to the characteristics of the unmanned vehicle, but for this reason, security vulnerabilities exist. A vulnerability in the MAVLink protocol can be used to send a command to a UAV pretending to be a GCS, and there is a study to neutralize it.

Disabling UAV can be divided into 3 steps according to the order of neutralization. First, the first step is to determine whether the UAV belonging to the neutralization target or the authenticated UAV is an identification friend or foe step of the UAV. This is a step that can be omitted in the No-drone Zone, such as military areas, hazardous facility areas, and privacy areas. The identification friend or foe step may include UAV identification through authentication key exchange or UAV identification through image processing using a camera

or sensor. The second step is to actually disable the UAV. This refers to the step of preventing the UAV from actually performing the assigned task through the aforementioned jamming, network attack, or physical attack. The third step is to safely lead the UAV to safe area so that the secondary damages are caused in the neutralization step, and the post-treatment is then secured. The reason why the third stage is important is that if UAV neutralization is carried out without the third stage, mainly in dangerous facilities such as nuclear facilities or densely populated areas, secondary damage through UAV crash can be caused. This can lead to serious property damage and human injury. In addition, neutralization through UAV hovering may not have secondary damage, but it is considered to be a perfect method of neutralization because it cannot be recovered. UAV disabling studies are generally studied in

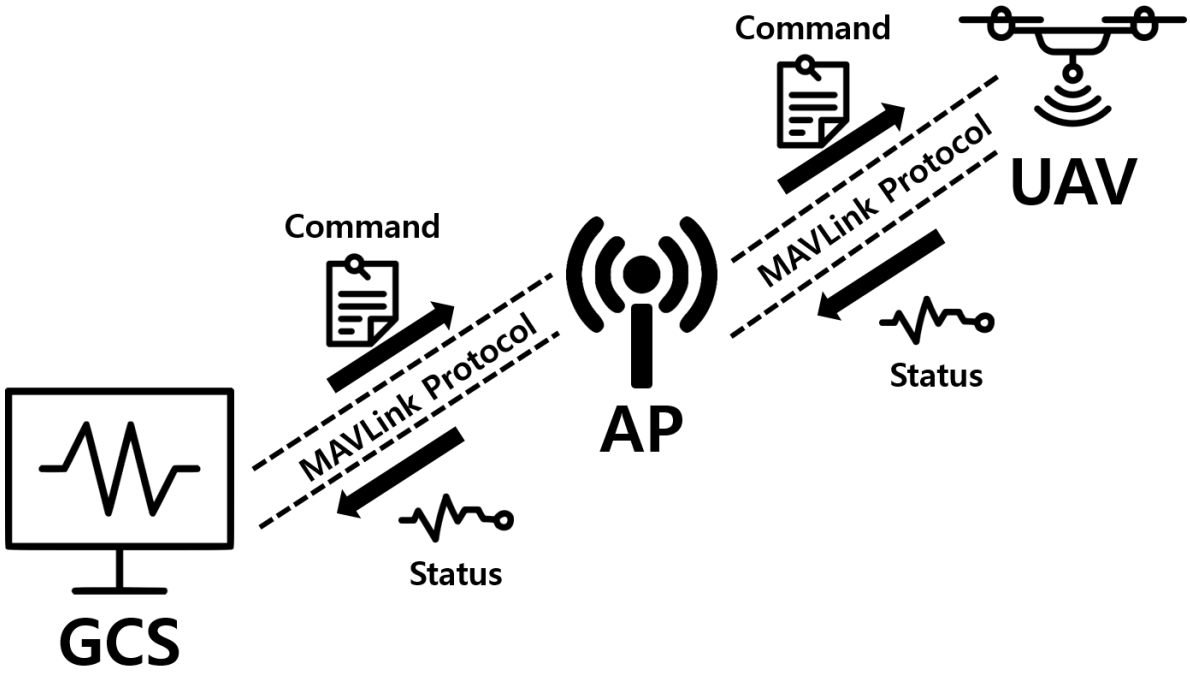


Figure 4. 1. UAS Structure.

the second stage, and the second stage, secondary damage, is usually not considered. In this paper, we would like to introduce the UAV neutralization method that was considered up to the third step.

In this paper, it is assumed that a UAV with ROS is applied. ROS provides various functions required for robots and has a large ecosystem. Therefore, it can be applied to various robots, sensors, and applications. In particular, in unmanned mobile vehicles, ROS is frequently used to develop functions for autonomous driving such as localization, navigation, mapping, and vision.

To use ROS, we applied MAVROS. MAVROS is one of the packages of ROS that is extended to use MAVLink protocol in unmanned vehicle using ROS. MAVROS can communicate with any unmanned vehicle that has MAVLink enabled. In addition, the MAVROS package provides communication drivers for autonomous flight using the MAVLink protocol, so it is mainly used for autonomous and military flights. When using MAVROS, the MAVLink 2.0 version can be used to support a large amount of data and IDs of 256 or higher when communicating using the ROS and MAVLink protocols. In this paper, MAVROS is applied for communication between PX4 flight stack and companion computer using ROS.

Unmanned vehicle mainly use MAVLink, a header-only message protocol optimized for this, because of their relatively limited resources. For this reason, PX4 and Ardupilot propose to use MAVROS, which provides a communication driver for various unmanned mobile vehicles along with the MAVLink protocol when using ROS.

It has been studied that ROS has a vulnerability to network attacks. The ROS can be divided into a publisher node and a subscriber node. At this time, since there is no function for identification or permission, unauthorized publishing or unauthorized subscription can be

Table 4. 1. CIA in ROS vulnerability.

Security Objective	ROS Vulnerability
Confidentiality	Unauthorized nodes can access ROS data
Integrity	Data published from unauthorized nodes
Availability	DoS attack due to the publication of a lot of fake data

Table 4. 2. Topics in MAVROS.

Plugin	Published topic	Subscribed topic
3dr_radio	~radio_status	-
actuator_control	-	~actuator_control
hil_controls	~hil_controls/hil_controls	-
global_position	~global_position/global ~global_position/local ….	-
imu_pub	~imu/data ~imu/data_raw …	-
manual_control	~manual_control/control	~manual_control/send
setpoint_position	-	~setpoint_position/global ~setpoint_position/local
waypoint	~mission/reached ~mission/waypoints	-
sys_time	~time_reference	-

performed through a packet injection attack. This can violate all three elements of information security, confidentiality, integrity and availability, also known as the CIA triad. Table 4.1 indicates this. An attacker could use this vulnerability to inject false data into the robot through unauthorized publishing, which could cause a fatal risk to the robot's behavior. In addition, the robot's status can be monitored through unauthorized subscriptions, and Denial of Service (DoS) attacks can be made by continuously posting fake data related to the robot's information processing or requiring any action.

MAVROS is also an extension package of ROS and inherits the vulnerability of ROS. However, the vulnerability of ROS to these network attacks is more vulnerable to unmanned mobile devices such as UAVs and poses a risk. MAVROS has a large number of topics for driving unmanned vehicles. The table 4.2 represents some of these topics. UAV's sensors and actuators exchange data through these topics. In this paper, we found that UAV can be disabled using the setpoint_position topic among these topics. This is an attack that violates the integrity of the CIA. Unlike other topics, the setpoint_position topic allows UAVs to subscribe to data published on this topic while on duty. Using this vulnerability, we attack via unauthorized data injection on the setpoint_position topic.

The figure 4.2 shows the attack situation briefly. Attackers attack when UAV and GCS using MAVROS and Pixhawk board are connected via Wi-Fi. When the attacker conducts attack with our proposed attack method, it seems that the UAV and GCS are injecting a command packet into the UAV while communicating. Inside the UAV, the attacker invades MAVROS of the UAV, creates and connects the publisher node of the attacker through the ROS master, and publishes data containing malicious values to the setpoint_position topic. Since the subscriber node cannot filter this, the impact of this attack eventually reaches the pixhawk board, and the UAV controls the UAV using the attacker's data.

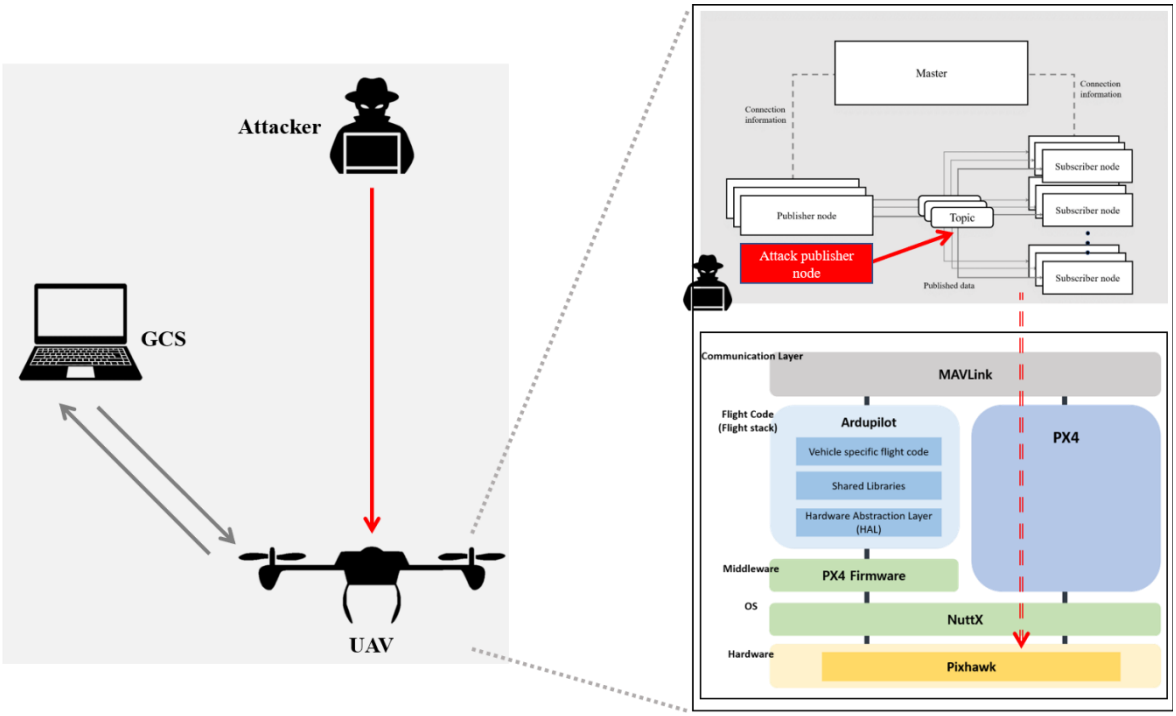


Figure 4. 2. The outline of the attack on the UAV using MAVROS.

4.2 Proposed defense method

We have confirmed that the above-mentioned vulnerability of MAVROS can land a UAV. This is an attack method that actually moves the UAV in a desired direction by injecting unauthorized data into the topic called setpoint_position. It is also possible to hijack a UAV to a desired location using this proposed attack method.

The attack method proposed in this paper is an unmanned vehicle neutralization method that corresponds to the second and third stages after performing the first identification friend or foe stage among the UAV disabling stages. In particular, our proposed attack method leads us to safely land on the ground regardless of the UAV's current altitude and landing point. This satisfies the third of the UAV neutralization phases. The use of this attack method has the advantage of preventing property damage and human injury that can occur due to the neutralization of the UAV, and safely and safely recovering the UAV without permission.

In many cases, unmanned vehicle performs tasks on behalf of people for tasks that are difficult for ordinary people to do directly or for tasks that are difficult for people to perform continuously. The attack method proposed in this paper can be a big vulnerability to unmanned vehicles that mainly use MAVROS to perform tasks such as autonomous driving and reconnaissance through mapping for the tasks of unmanned vehicles.

To prevent this vulnerability of MAVROS, this paper proposes a security API for unmanned vehicles. API for security is an API that applies on top of the ROS system without modifying or deleting unique tasks such as ROS process processing and data processing. This does not affect the behavior of the robot even if the security API we proposed is uploaded in an application using ROS.

An unmanned mobile vehicle that takes the place of an ordinary person starts a mission and measures data in real time using a mounted sensor. In addition, information is processed using

the measured data, and control values for missions are generated. The mounted actuators perform the task by receiving the generated control values. These unmanned moving objects usually do not connect any sensors or actuators other than the ones or actuators already installed when the mission starts. In this paper, we introduce the defense API against attacks using MAVROS vulnerabilities, which are proposed assuming the characteristics of unmanned vehicles.

First, the connection process between nodes in the general MAVROS will be described. The figure 4.3 briefly describes the connection between nodes in ROS including MAVROS. Assuming that there are ROS master, publisher node, and subscriber node, ros master receives information necessary for connection from subscriber node and publisher node that

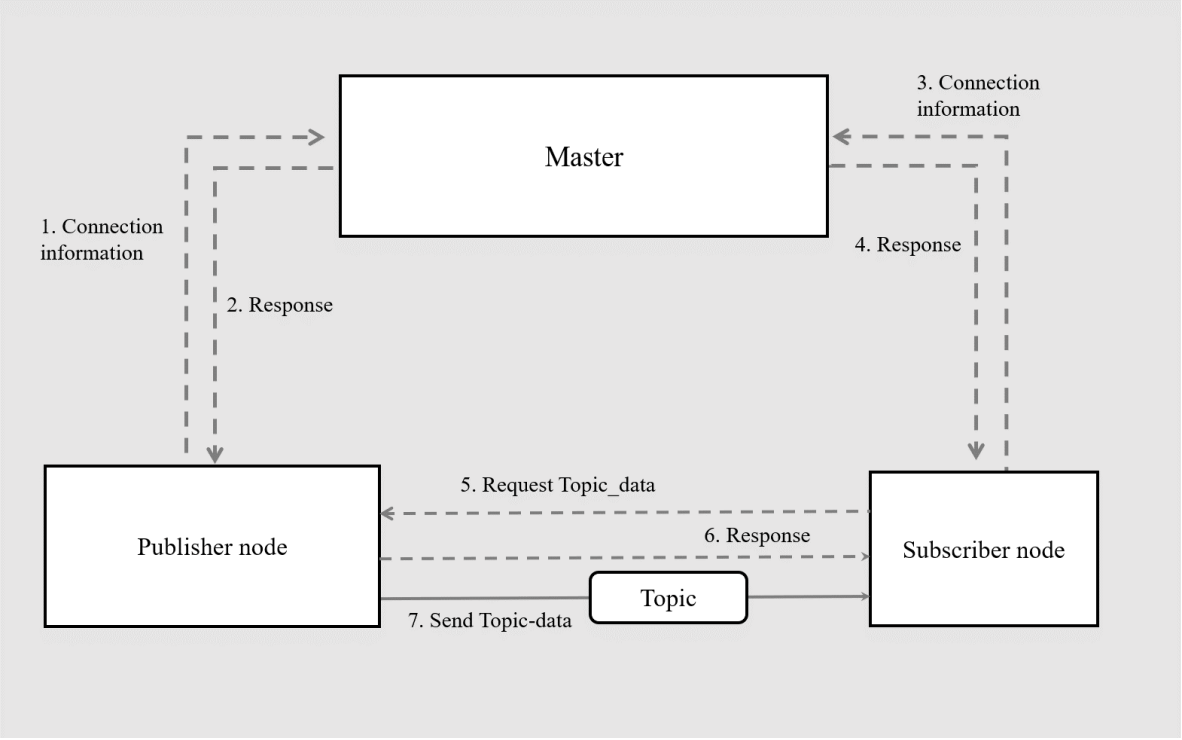


Figure 4. 3. Connection process in ROS.

want to connect using XML-RPC protocol, which is an HTTP-based protocol. At this time, the nodes transmit the topic name, node name, URI address and port information to the ROS master. Thereafter, communication between nodes receives a message for the topic through registered topic information. Looking at it in order, the publisher node sends connection information such as topic name and publisher node URI to the master, and when a response is received from the master, the publisher node is registered to the topic. Next, the subscriber node also sends connection information such as topic name and subscriber node URI, and receives and registers information about the publisher node connected to the topic with a response from the master. After that, the subscriber node requests topic data from the publisher node registered in the topic and receives a response from the publisher node. After going through this series of processes, the publisher node and the subscriber node are connected and communicate directly without the ros master. At this time, TCP or UDP can be set. Logically, topics can be viewed as large message buses. In one MAVROS, the number of nodes can be exist. Each node is recommended to perform only one function, and one node can be connected to multiple topics.

In the process of connecting nodes in ROS including MAVROS, the ros master connects unauthorized nodes to normal and authorized nodes through topic registration without a series of verification processes. In this process, if the network has been previously intruded, unauthorized nodes can be registered through the ros master through this vulnerability.

The security API proposed in this paper is a method to prevent the registration of unauthorized nodes. The proposed method prevents node registration when the unmanned moving object completes the installation of sensors and actuators and then starts a mission such as flight, driving, or navigation, or when the installation of sensors and actuators no longer occurs. That is, in the normal situation, the absence of mounting of sensors and

actuators assumes that there is no additional node registration. At this time, a normal user can lock the MAVROS application, and the locked application can no longer register the node. The lock mode and unlock mode can be set dynamically by the user, and the lock mode of the MAVROS application can be set using the identification code previously entered by the user.

For example, when a UAV performing a mission starts flying, the user locks the UAV's MAVROS through a identification number. In this case, the attacker cannot register the unauthorized node in the UAV's MAVROS. In addition, if you want to convert the lock mode UAV to unlock mode, the user can convert it to unlock mode using a identification number.

V. SIMULATION RESULT

5.1 Experiment environment for attack and defense on UAV simulation

In this paper, we set up an experimental environment to run our proposed attack simulation. The figure 5.1 briefly shows the experimental environment we have constructed. UAS is composed of UAV and GCS connected via Wi-Fi. Pixhawk2 is connected to the UAV, and a flight stack is applied to this board to perform the task for flight. In addition, the Pixhawk board operates as a communication layer including MAVLink. Also, Raspberry pi board is installed as companion board to install MAVROS in UAV. MAVROS is applied to the Raspberry pi and it is connected to the pixhawk board via Wi-Fi.

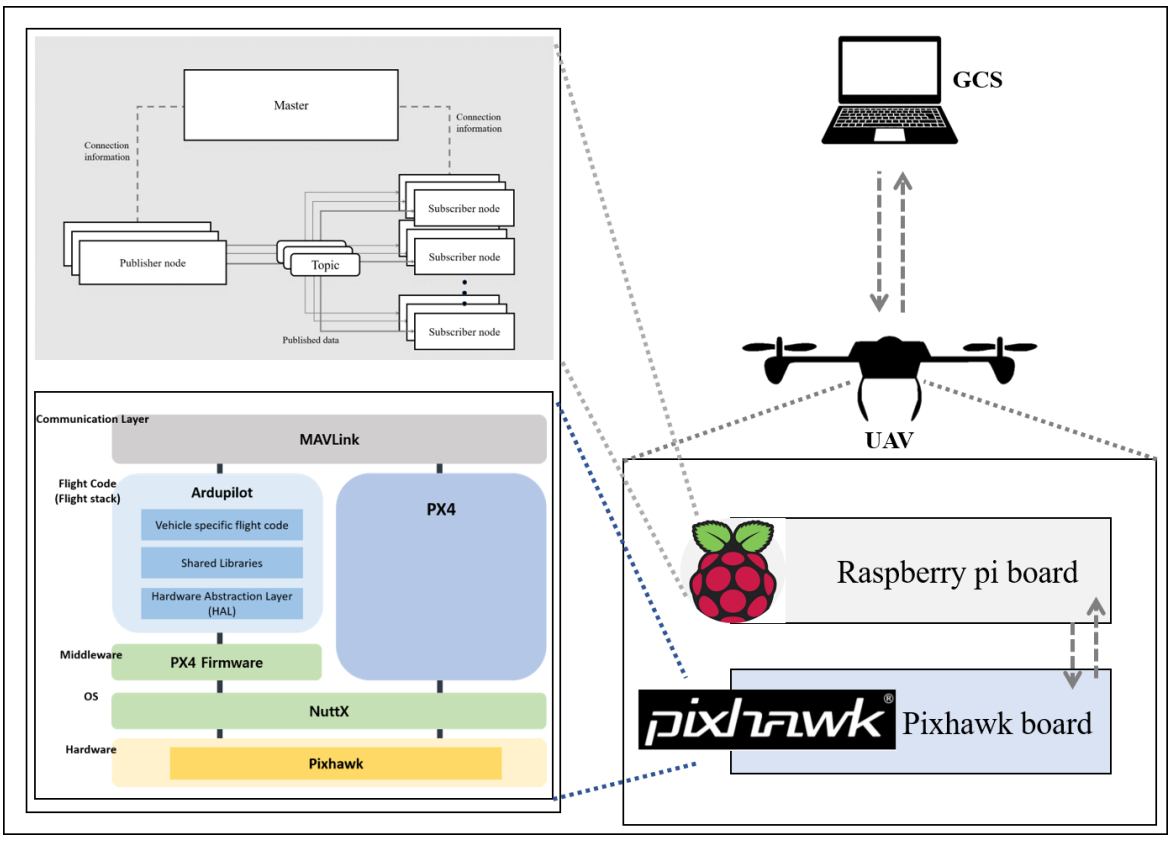


Figure 5. 1. UAS structure using MAVROS.

The experimental environment prepared is almost the same as the actual UAV environment. The actual UAV also has an ESP8266 that acts as an AP on the Pixhawk board, enabling Wi-Fi communication. At this time, the MavESP8266 firmware is mainly used, and the GCS and Wi-Fi are connected using the MAVLink protocol. And to use MAVROS, Raspberry pi is installed. Pixhawk board and Raspberry pi board are connected by wire or wirelessly. The simulation environment is also the same as a real UAV, and the Pixhawk board and Raspberry pi board are connected wirelessly. Before, running the attack simulation, we will look at the flight process in a normal situation without an attacker. UAV with MAVROS applied will fly in OFFBOARD_MODE. This means that the flight will be made with the source code embedded in the Raspberry pi board. In normal case, first run ros core on Raspberry pi board and execute ros launch command to connect flight source code to MAVROS. And in UAV, arm means ready to fly. When the UAV is ready for flight, when the ros run command is executed, it will fly according to the flight source code, which is a built-in code.

Let's take a look at the flight preparation process in the attack simulation. The attacker is added during the normal flight preparation process. It is assumed that the attacker has previously entered the UAV's network. An attacker who entered the network first finds the address of the ROS master through port scan. The attacker who finds the URI of the ros master connects to the external ros master through the export ROS_MASTER_URI command. In the simulation in this paper, it connects to theros master of the external UAV through the command export ROS_MASTER_URI = http://192.168.1.117:1311. In this way, an attacker connected to the ROS master of UAV's MAVROS injects fake data into setpoint_position among the UAV's MAVROS topics.

We used the setpoint_position of MAVROS to satisfy the third phase of the UAV disabling phase. Strictly speaking, we intend to disable the UAV by satisfying the third stage by using

two setpoint_position/local and setpoint_position/global in the setpoint_position, which is a plug-in of MAVROS. We have found that an attacker can use the setpoint_position/local topic to switch the UAV to Return To Launch (RTL) mode to return to the starting point. The way to change UAV to RTL mode is as follows. In the Setpoint_position topic, you can specify the x, y, and z coordinates of the UAV, and the first takeoff point becomes (0, 0, 0). If you post (0, 0, 0) when posting data on this topic, the UAV will fly to the first take-off point. In other words, if an attacker injects fake data by setting x, y, and z values to (0, 0, 0) on the setpoint_position/local topic, the UAV returns to the takeoff point as if the mode was changed to RTL mode.

In the case of the attack method using the setpoint_position/global topic in the setpoint_position plug-in of MAVROS, the attacker can hijack the UAV to the desired point. In the case of setpoint_position/local, the first takeoff point is set to (0, 0, 0), and the UAV is routed using the relative address, whereas setpoint_position/global can route the UAV using GPS coordinates. Using our proposed attack method, this topic can also be easily generated and injected by an attacker by connecting to UAV's MAVROS as a publisher node.

5.2 Simulation result for attack on UAV

First, in the normal case, the attack methods using the setpoint_position/local and setpoint_position/global topics are compared. In the normal case, it means that the UAV is connected to the GCS and performs the task of flying along a predetermined route. To compare it, our proposed attack method is to inject the attack data with the setpoint_position/local topic while the UAV is connected to the GCS and flying along a predetermined path. At this point, the attacker sets the UAV's target x, y, and z values to (0, 0, 0) to return to the point where the UAV first took off. Lastly, an attacker injects attack data with the setpoint_position/global topic. At this time, the attacker sets the target GPS value of the UAV separately to move the UAV to the desired location.

The figure 5.2 shows the case where the attack value is injected into the setpoint_position/local and setpoint_position/global topics in the normal case through the UAV flight path. Among the relative coordinates of the UAV, the x-axis represents the x-value of the UAV, and the y-axis represents the y-value of the UAV. Normally, it is a flight path that takes off from the (0, 0) starting point and lands at about (170, 260), but if the attacker injects an attack value into the setpoint_position/local topic, from the attack point at about (90, 210) You can see that the UAV returns to the (0, 0) takeoff point again. In addition, when the attacker injects an attack value into the setpoint_position/global topic, it can be confirmed that the UAV has moved from the attack point of about (110, 220) to the GPS injected by the attacker rather than the original mission destination. That is, it can be confirmed that the attacker can return the UAV to the takeoff point or move the UAV to the desired GPS coordinates if the attack method proposed in this paper is used despite the original mission destination of the UAV.

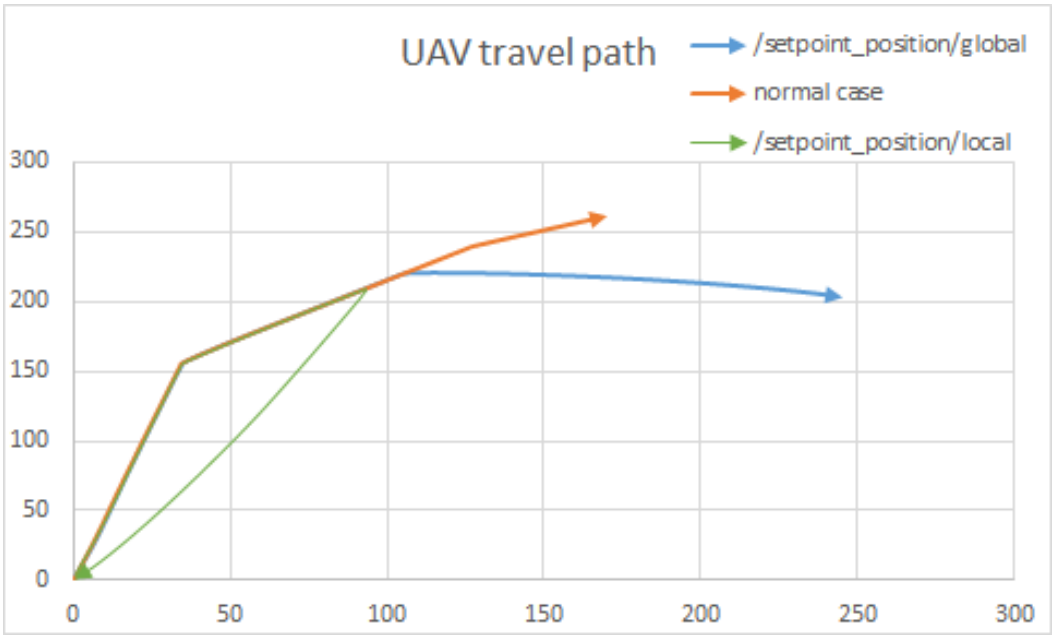


Figure 5. 2. UAV travel path for each situation.

We make sure that our proposed attack method satisfies the third of the aforementioned UAV neutralization phases. In order to compare this, the third step of the unmanned neutralization phase is compared with the neutralization method. The neutralization method that does not satisfy the third step is a method of neutralizing the UAV using the vulnerability of the MAVLink protocol, and the attacker injects a disarm packet while the UAV is performing the task. The UAV attacked in this way stops the wing from spinning and crashes at the altitude of the flight, even though it is in flight without being able to filter out the disarm packet sent by the attacker. In this case, fatal property damage and human injury may occur depending on the local situation and the UAV flight altitude situation. With this attack method, not only disarm packet but also mission_count packet is possible.

We compared the packet injection attack, which is an attack method that does not satisfy the third step, and the attack using the vulnerability of MAVROS, our proposed attack method. First, the flight was assigned to the UAV of the simulation to make the flight, and the attacker attacked at 70 and 60 seconds respectively. The figure 5.3 shows the situation

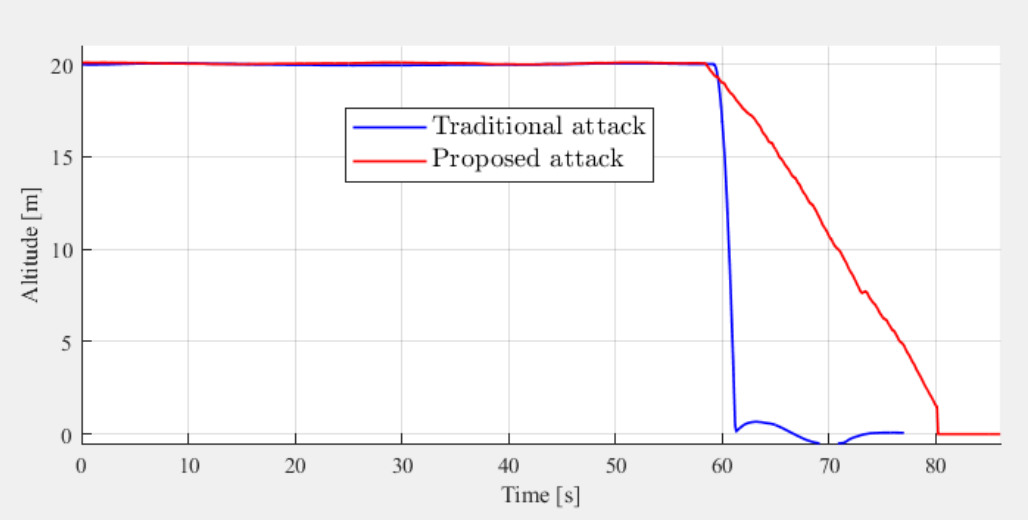
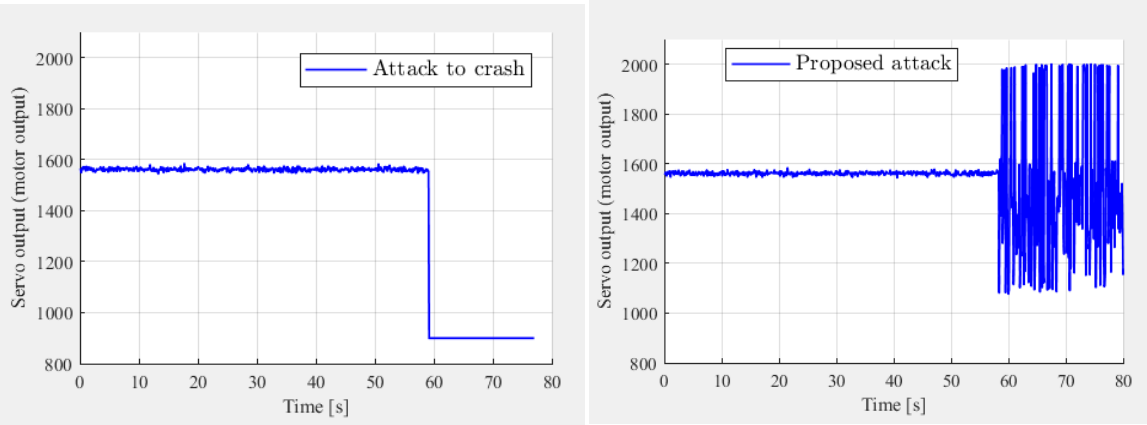


Figure 5. 3. UAV altitude in traditional and proposed attack.

where the attacker attacked during UAV's flight mission through altitude. In Figure 5.3, blue line shows an attack through a vulnerability in the MAVLink protocol, where the UAV sent a disarm packet during a flight mission for disabling the UAV. Also, you can see that the UAV that was flying while maintaining the altitude at 20 dropped to zero and hit the ground in less than 1 second. On the other hand, Red line shows the UAV altitude when we use the proposed attack method, and the UAV, which was well maintained at the altitude of 20, gradually lowers the altitude and lands. Through this, it can be seen that it can be neutralized in a safer way, rather than neutralized through the fall of the UAV.

Also, it is possible to compare the altitude through the output coefficient of the motor through the figure 5.4. The attack using the vulnerability of the MAVLink protocol is briefly shown in Figure 5.4 (a). The servo_output was 1600, and the UAV was flying, but the servo_output value, which is the value at which the rotation of the wing stops as soon as the attacker attacks in about 70 seconds, is 900. Through this, it can be checked that the UAV stops rotating the wing, even though it is flying at a high altitude, that is, it becomes



(a) (b)

Figure 5. 4. UAV servo output in traditional and proposed attack.

neutralized through a fall. However, Figure 5.4 (b) shows the case of attacking UAV using our proposed attack method. The UAV was flying with a servo_output value of about 1600, but it can be seen that the servo_output value does not drop abruptly after the attack point in about 60 seconds, but the value changes rapidly. Because, in this process, UAV received original control message and attacker message at the same time. Figure 5.5 shows the SITL under attack using setpoint_position/global. Through this, we confirmed that the UAV landed on the ground in a safe state even if it takes a little time.

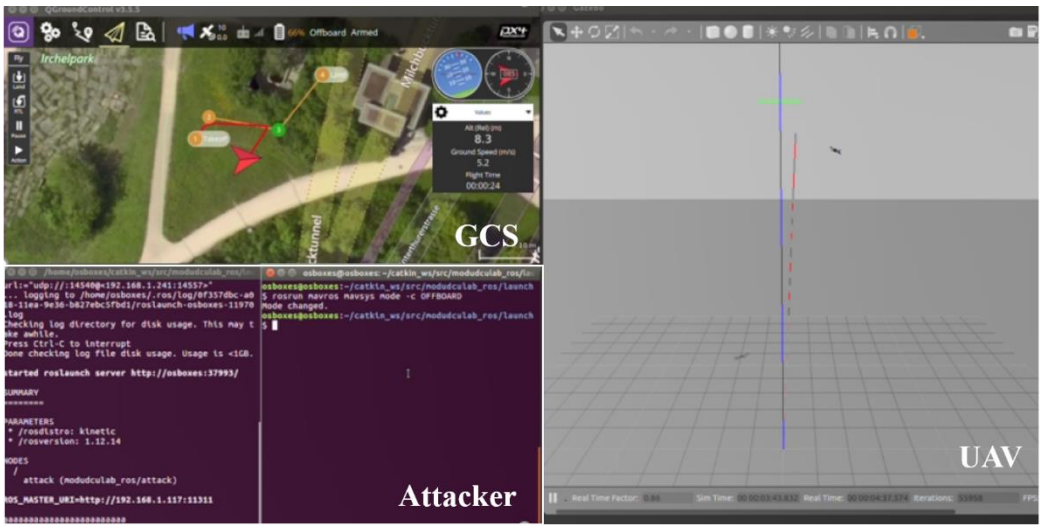


Figure 5. 5. SITL in attack using setpoint_position/global.

5.2 Simulation result for defense on UAV

The simulation confirmed that the proposed attack method is applied to UAV and MAVROS. In particular, it was confirmed that an attacker publishes unauthorized data as the publisher node with the relative and absolute coordinates of the UAV on the setpoint_position / local and setpoint_position / global topics, and the UAV with MAVROS applied does not filter it and accepts the unauthorized data as it is. This can be a vulnerability that poses a tremendous risk for unmanned mobile vehicles that perform tasks that are difficult or difficult for humans to perform along with tasks such as flight, driving, and navigation. Therefore, defense measures are needed.

In this paper, we introduce the API for security in MAVROS. The API for security that we propose keeps the existing MAVROS operation intact, but only when a new publisher node is connected by the ros master determines whether to connect the new publisher node through some process check. This makes it possible to take advantage of the characteristics of an unmanned mobile vehicle that rarely connects to a new publisher node or subscriber node when performing a mission. We put a lock mode on the ros master so that new publisher

Algorithm 1. The defense algorithm

```
1:  if node.name is UNLOCK
2:      release the lock
3:  if node registration is not available
4:      exit the process
5:  if node.name is LOCK
6:      acquire the lock
7:  registration procedure
```

nodes cannot be connected. This allows a normal user to dynamically lock or unlock the mode.

The algorithm 1 represents a series of processes that must be performed whenever a new publisher node requests a connection in MAVROS to which an API for security is applied. UAV using MAVROS with the API for security proposed by us can be divided into three states by the security API. The figure 5.5 is a flow diagram showing the state of UAV. First, the UAV starts in the idle state, not in both lock mode and unlock mode. In this state, any publisher node or subscriber node can be registered in the UAV's MAVROS. Usually, the UAV is in preparation before starting the mission. When entering the lock state from the idle state, no publisher node can be registered. This state is used when the UAV starts flying and performs a mission, or when the network is vulnerable. And while the UAV is flying, a normal user can reach the unlock state by entering a identification number. In the unlock state, new publisher nodes can be registered. In this way, only the situational management of the UAV to which MAVROS is applied can prevent the most vulnerable point of MAVROS.

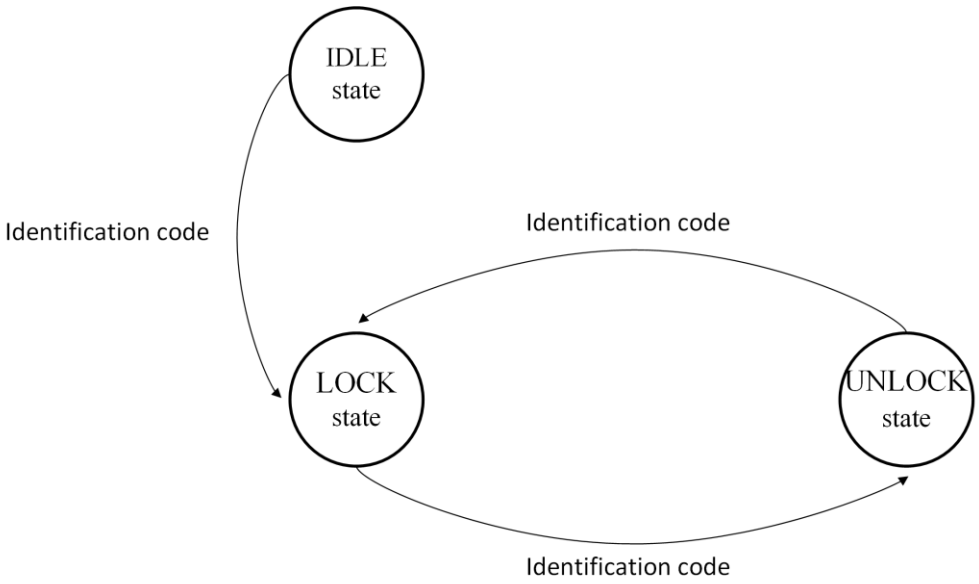


Figure 5. 6. UAV state machine in defense API.

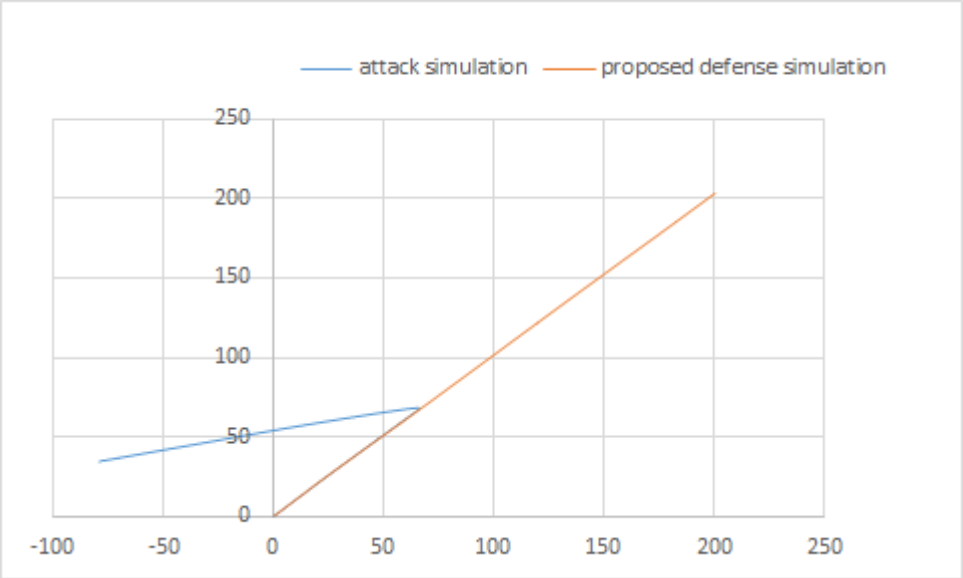


Figure 5. 7. Flight path in existing and proposed defense system.

When the API for security proposed by us was applied to MAVROS, the proposed attack method was implemented. The figure 5.6 shows the flight path of the UAV. At this time, the x-axis and y-axis of the graph represent relative coordinates based on the takeoff point (0, 0) of the UAV. This is the case when the target point is designated as (100, 100) in the (0, 0) coordinate which is the takeoff point to UAV. First, if the API for security is not applied, it can be checked that the attacker is attacked at about (60, 60) points and fails to perform the task that it was originally trying to perform, but moves to the coordinates designated by the attacker. UAV with our proposed security API is applied ignores the attack value because the publisher node of the attacker cannot be registered. Therefore, it can be seen that the UAV's mission is fully fulfilled.

In addition, you can check the use of the security API through altitude. If you look at the figure 5.7, if you look at the case of MAVROS where the proposed defense API is not applied, you can see that it is attacked and descends at an altitude of 10. This is the implementation of our proposed attack method, but it doesn't crash, but fails to succeed and descends and lands eventually. However, although the UAV of MAVROS with the proposed defense API

was attacked at the same altitude of 10, it can be seen that the UAV in the lock state successfully ignores the attacker's attack and maintains an altitude of 20.

Figure 5.5 shows the SITL under attack on system applied proposed defense API. Through this, we confirmed that the UAV successfully complete its mission despite attack.

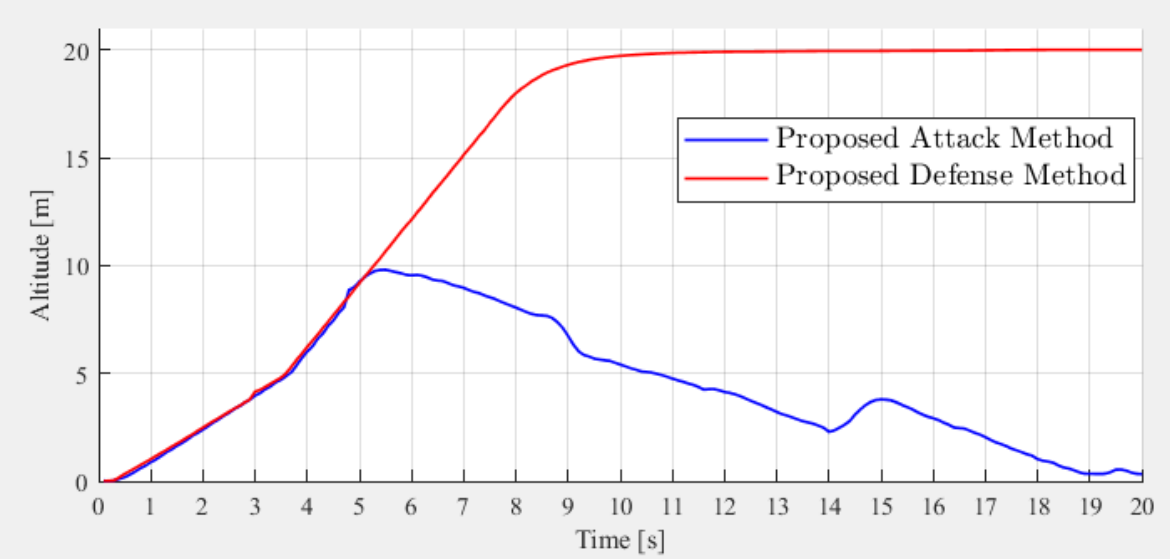


Figure 5. 8. UAV altitude in existing and proposed defense system.

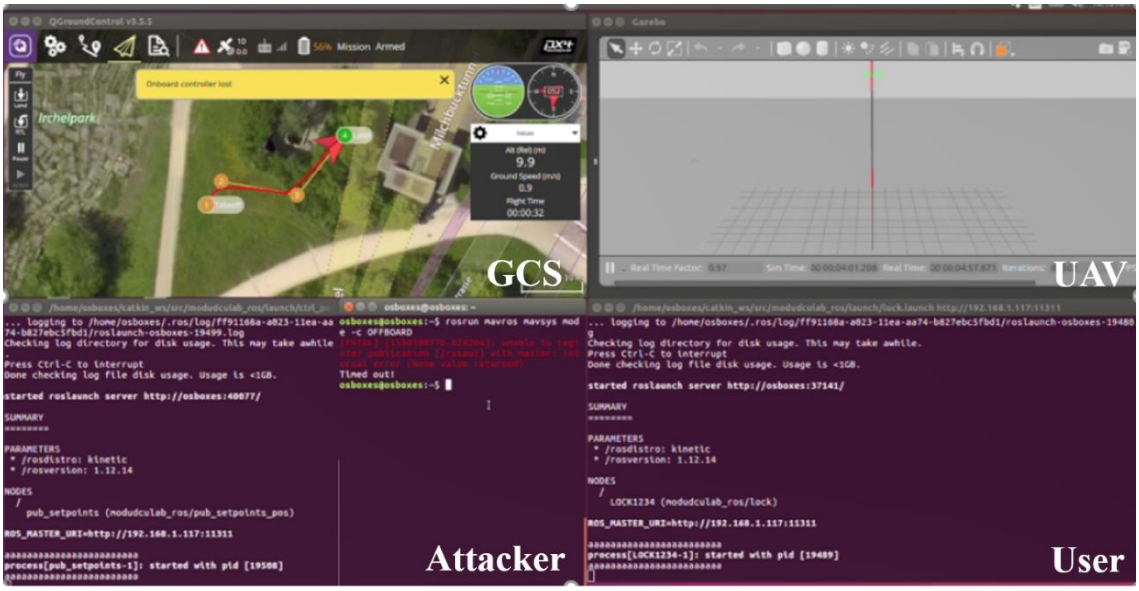


Figure 5. 9. SITL in attack simulation on system applied proposed defense API.

VI. CONCLUSION

As the unmanned vehicle, one of the main applications of CPS, expands its use, research on the neutralization of the unmanned vehicle is also becoming important. We assumed UAV among unmanned vehicles. The UAV neutralization phase can be divided into three phases: the UAV identification phase, UAV mission neutralization phase, UAV safety recovery and post-processing phase. However, most studies do not consider the UAV safety recovery and post-processing step, and are only in the process of neutralizing the UAV mission. We have found a vulnerability in the commonly used MAVROS environment to implement functions such as autonomous driving, navigation, and collision avoidance, and proposed an attack method using the vulnerability. The proposed attack method doesn't cause secondary damage by landing to a desired point or returning it to a take-off point, not an UAV disabling through a fall. Also, we proposed MAVROS API for light security to prevent this vulnerability. This was solved by dividing the UAV state into three states: IDLE, LOCK, and UNLOCK to prevent the registration of new publisher nodes.

References

[1] K.-J. Park, R. Zheng, and X. Liu, "*Cyber-physical systems: Milestones and research challenges*," *Computer Communications*, 36(1), 2012, pp. 1-7.

[2] K.-J. Park, J. Kim, H. Lim, and Y. Eun, "*Robust path diversity for network quality of service in cyber-physical systems*," *IEEE Transactions on Industrial Informatics*, 10(4), 2014, pp. 2204-2215.

[3] Y.-M. Kwon, et al., *"Empirical analysis of mavlink protocol vulnerability for attacking unmanned aerial vehicles," IEEE Access*, vol. 6, pp. 43203-43212, 2018.

[4] J. Y. Yoon, H. J. Lee, and K. J. Park, *"Security enhancement in wi-fi communication between UAV and GCS," in Proc. KICS ICC 2019*, pp. 400-401, Jun. 2019.

[5] J. M. YU, J. Y. Yoon, and K. J. Park. *"Risk analysis of UAV and GCS for network attacks," J. KIISE*, vol. 37, no. 1, pp. 29-37, Jan. 2019.

[6] A. Y. Javaid, et al., *"Cyber security threat analysis and modeling of an unmanned aerial vehicle system," 2012 IEEE Conf. Technol. for Homeland Secur. (HST),* Waltham, MA, USA, Nov. 2012.

[7] J. Y. Yoon, J. M. Yu, and K. J. Park. *" UAV Network Security Enhancement Model Using Private Blockchain," in Proc. KICS Int. Conf. Commun. 2019 (KICS ICC 2019),* pp. 1259-1260, yong pyong, Korea, January 2019.

[8] S. Deng, et al., *"Packet injection attack and its defense in software-defined networks," IEEE Trans. Inf. Forensics and Secur.,* vol. 13, no. 3, pp. 695-705, 2017.

[9] E. Bertino and N. Islam, *"Botnets and internet of things security," Computer,* vol. 50, no. 2, pp. 76-79, Feb. 2017.

[10] A. K. Simpson, F. Roesner, and T. Kohno, *"Securing vulnerable home IoT devices with an in-hub security manager," 2017 IEEE PerCom Workshops*, Kona, HI, USA, Mar. 2017

[11]Bhattacharya, Sourabh, and Tamer Başar. "Game-theoretic analysis of an aerial jamming attack on a UAV communication network." *Proceedings of the 2010 American Control Conference*. IEEE, 2010.

[12]Sliti, Maha, Walid Abdallah, and Noureddine Boudriga. "Jamming Attack Detection in Optical UAV Networks." *2018 20th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2018.

[13] Dieber, Bernhard, et al. *"Security for the robot operating system." Robotics and Autonomous Systems 98 (2017)*: 192-203.

[14] Breiling, Benjamin, Bernhard Dieber, and Peter Schartner. *"Secure communication for the robot operating system." 2017 annual IEEE international systems conference (SysCon).* IEEE, 2017.

[15] Mukhandi, Munkenyi, et al. *"A novel solution for securing robot communications based on the MQTT protocol and ROS." 2019 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2019.

[16] Lera, Francisco Javier Rodrıguez, et al. *"Cybersecurity in Autonomous Systems: Evaluating the performance of hardening ROS." Málaga,* Spain 47 (2016).

[17] Dóczi, Roland, et al. *"Increasing ros 1. x communication security for medical surgery robot." 2016 IEEE International Conference on Systems*, Man, and Cybernetics (SMC). IEEE, 2016.

[18] Dieber, Bernhard, et al. *"Application-level security for ROS-based applications." 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016.

[19] Huang, Jeff, et al. *"ROSRV: Runtime verification for robots." International Conference on Runtime Verification*. Springer, Cham, 2014.

[20] Rivera, Sean, Sofiane Lagraa, and Radu State. *"ROSploit: Cybersecurity tool for ROS." 2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2019.

요 약 문

MAVROS를 사용하는 무인이동체에서의 공격과 방어 실증 연구

최근 CPS의 주요 어플리케이션 중 하나인 무인 이동체가 용도를 넓힘에 따라 무인 이동체의 무력화 연구도 중요해지고 있다. 우리는 무인 이동체 중 UAV를 이용하였다. 무인기 무력화 단계는 무인기 피아 식별 단계, 무인기 임무 무력화 단계, 무인기 안전 회수 및 사후 처리단계로 크게 세 단계로 나뉠 수 있다. 그러나 대부분의 연구는 무인기 안전 회수 및 사후 처리단계까지 고려하지 않고 있으며 무인기 임무 무력화 단계에 그치고 있다. 무인기 안전 회수 및 사후 처리단계까지 고려하지 않은 무력화는 일반적으로 추락을 통한 무력화, 호버링을 통한 무력화가 있다. 이는 무력화 단계에서 심각한 재산, 인명 등의 2차 피해를 초래할 수 있다.

본 논문에서는 자율 주행, navigation, 충돌 회피 등의 기능 구현을 위해 무인 이동체에서 일반적으로 사용하는 MAVROS 환경의 취약점을 찾고 그 취약점을 이용하여 공격방법을 제안하였다. 제안하는 공격방법은 추락을 통한 무인기 무력화가 아닌 원하는 지점으로 착륙시키거나 이륙지점으로 돌려보내는 무력화 방법으로 이로 인한 2차피해가 발생하지 않는다. 특히 원하는 지점으로 착륙시키는 공격 방법은 무력화 대상인 UAV를 탈취할 수 있다는 점이 있다.

이러한 취약점을 막기 위해 우리는 또한 보안을 위한 MAVROS API를 제안하였다. 이는 UAV의 상태를 IDLE, LOCK, UNLOCK 세가지 상태로 나누어 동작한다. MAVROS에 새로운 publisher node가 등록될 때 몇 가지 일련의 검사과정을 거치며 사용자가 동적으로 고유번호를 이용하여 상태를 변화시킬 수 있다는 점이 있다. 실험을 통해 공격자의 publisher node 등록을 막음으로써 공격을 막는 것을 확인할 수 있었다.

핵심어: Network attack, Unmanned Aerial Vehicle (UAV), Security, ROS, MAVROS,