OTAC review task

드론 보안성 분석 보고서

September, 2021

고려대학교 정보보호대학원
무인이동체보안연구센터

KOREA UNIVERSITY

고려대학교 정보보호대학원
Korea University
School of Cybersecurity

Table of contents

List of tables

List of figures

1 Drone network protocol 분석

1.1 MAVLink 개요

STX (0xFD)	LEN	INC FLAGS	CMP FLAGS	SEQ	SYS ID	COMP ID	MSG ID 3 bytes	PAYLOAD 0–255 bytes	CHECKSUM 2 bytes

Figure 1.1: Structure of MAVLink 2.0 frame

MAVLink (Micro Air Vehicle Link)는 지상 통제 시스템(GCS; Ground Control System)과 비행체 간의 통신 혹은 비행체 내부 장치 간 통신을 위해 사용되는 메세징 프로토콜이다. Figure 1.1은 MAVLink 2.0의 프레임 구조를 나타낸다. 노드들은 MSG ID를 기준으로 PAYLOAD를 생성 또는 해석하며, 기본적인 드론 운용을 위한 다양한 메세지 타입이 미리 정의되어있다. MAVLink 설계 시 보안에 대한 고려가 되어있지 않아 공격자가 쉽게 메세지 위/변조할 수 있다.

1.2 무인이동체 내부 네트워크

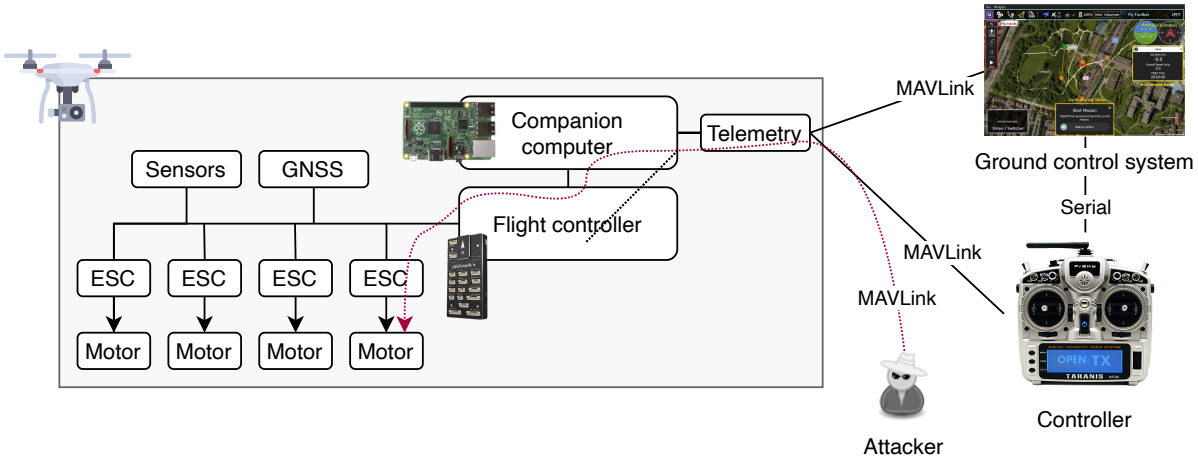


Figure 1.2: Drone network components and network topologies

Figure 1.2는 무인이동체 네트워크 구성도를 나타낸다. 무인이동체 네트워크는 드론 본체, 지상 통제 시스템, 조종기로 구성되며 일반적으로 드론은 GCS 또는 조종기에 의해 제어된다. 각 노드는 MAVLink 프로토콜을 기반으로 서로 통신한다. 지상 통제 시스템과 컨트롤러의 차이점은 다음과 같다.

GCS GCS는 드론 모니터링, Way-point 기반의 드론 제어, 드론 환경설정 등 다양한 목적으로 사용된다. GCS는 드론이 주기적으로 전송하는 정보 (지리적 위치, 고도, 자세 정보, 배터리 잔량 등)를 해석하고 이를 사람이 읽기 쉬운 형태로 구성하여 화면에 도시한다. 운전자는 GCS를 활용하여 특정 Way-point를 미리 드론에 입력해 두고 조종기 없이 드론이 스스로 임무를 수행하는 환경을 구성할 수도 있다.

조종기 조종기는 드론의 움직임을 직접 제어하기 위해 사용된다. 일반적으로 조종기에는 throttle, yaw, pitch, raw 4개의 채널에 해당하는 조이스틱이 존재하며, 운전자가 제어하는 조이스틱의 값은 PWM 신호의 형태로 드론에 전송된다.

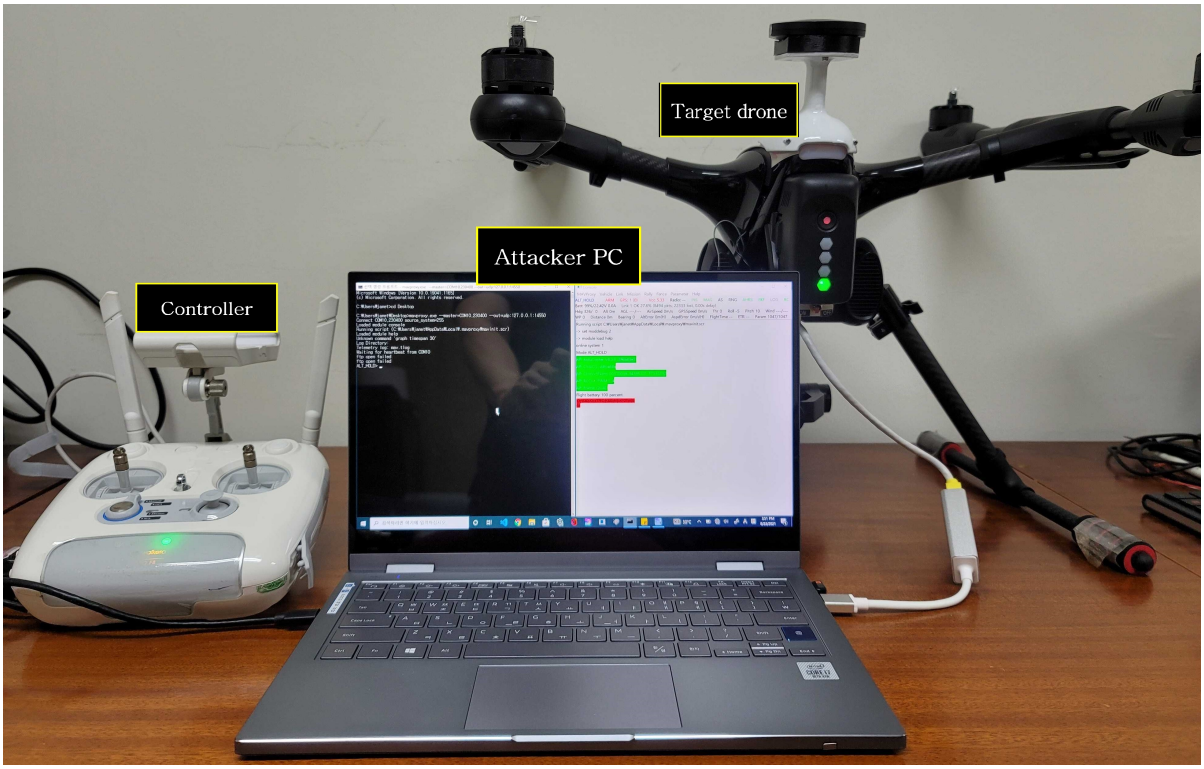


Figure 1.3: Experimental setup

1.3 GCS 또는 조종기로 위장한 공격자

Figure 1.3은 공격자를 포함한 실험 환경 구성을 나타낸다. 드론 네트워크에 침투한 공격자는 네트워크 노드들(드론, GCS, 조종기 등)이 생성하는 MAVLink 메세지를 스니핑하거나 직접 MAVLink 메세지를 생성하여 특정 노드에게 전송할 수 있다. 본 연구에서는 드론 네트워크에 침투한 공격자를 구현하기 위해 공격자 PC에서 MAVProxy[1]를 통해 조종기를 제어하였다. Python 라이브러리인 pymavlink[2]를 이용하여 공격 메세지를 구현하고 이를 조종기에 전달하여 드론에 공격 메세지를 전송하였다. 공격의 유형과 공격으로 인한 효과는 아래와 같다.

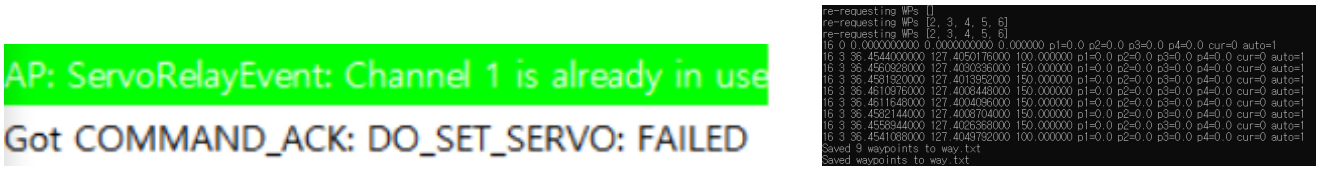
ARM/DISARM `MAV_CMD_COMPONENT_ARM_DISARM` 명령어를 주입하여 임의로 드론에 시동을 걸거나 시동을 끌 수 있었다.

조종기 채널 PWM 값 주입 공격 공격자가 채널 PWM 값을 임의로 생성하여 이를 드론에 주입하였다. Figure 1.4a는 공격자 PC에서 채널 PWM 값 주입 공격을 시도하는 모습을 보여준다. 드론은 공격 메세지를 정상적으로 수신하였으나 본 실험 환경에서는 이미 조종기가 모든 채널을 점유하고 있었기 때문에 실질적으로 공격의 효과가 나타나지는 않았다.

미션 조작 공격 드론에 등록된 Way-point를 추출하거나 임의의 Way-point를 삭제할 수 있었다. Figure 1.4b는 공격자 PC에서 확인한 드론의 Way-point를 나타낸다.

서비스 거부 공격 `HEARTBEAT`, `PING`, `PARAM_REQUEST_LIST` 등의 MAVLink command를 과도하게 전송하여 드론 내부에 비정상적인 부하를 일으킬 수 있었다.

[1] https://ardupilot.org/mavproxy/
[2] https://github.com/ArduPilot/pymavlink



(a) Channel PWM injection attack (b) Way-point disclosure attack

Figure 1.4: Simulation results of the attacks.

1.4 정리

드론 네트워크에 침투하기만 한다면 공격자는 Remocopter 500 드론를 통제할 수 있는 다양한 수단을 얻을 수 있다는 사실을 확인했다. 실제 실험을 통해 드론의 시동을 끄고 켜는 공격, Way-point를 조작하는 공격, 드론 내부에 비정상적인 부하를 발생시키는 공격 등이 가능함을 입증했다.

2 OTAC 안전성 분석

2.1 OTAC 패킷 분석

2.1.1 패킷 분석

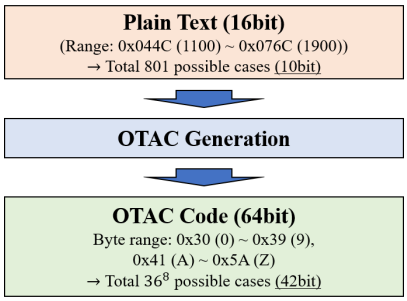
OTAC 코드생성 알고리즘은 16비트 길이의 평문을 80비트 길이의 OTAC 코드로 변환한다. 평문 데이터는 드론의 비행제어에 사용되는 명령어에 해당하며, 80비트의 OTAC 코드중 2바이트는 0x0000 으로 고정값을 가지고 있다. 또한, 제공된 시리얼 모니터링 툴을 활용하여 (평문, 코드)쌍을 분석한 결과, 조종기의 명령은 4개의 채널을 통해 전송되며, 모든 채널은 공통된 seed를 사용하여 OTAC 코드를 생성하는 것을 확인하였다. 다시 말해, OTAC 코드 변환 주기 내에서 평문 M에 대응되는 OTAC 코드 C는 4개의 채널에서 모두 동일하였다. 또한, OTAC 코드 변환 주기 내에서는 평문 M에 대응되는 OTAC 코드 C는 고정값을 가지는 것을 확인하였다. 하지만, 짧은 업데이트 주기로 인해 공격자는 획득한 OTAC 코드를 이용하여 유효한 공격을 수행하기 어렵다. Figure 2.1a은 평문과 이에 대응하는 OTAC 코드를 보여준다.

드론 제어명령인 평문은 십진수 1100 부터 1900까지의 범위를 가지며, 이는 총 801가지 경우의 수로 표현된다.

OTAC 코드의 개별 바이트 값은 ASCII 코드에서 0 ~ 9 범위의 십진수와 A ~ Z 범위의 대문자 알파벳을 사용한다. 따라서 OTAC 코드의 개별 바이트는 36가지 경우의 수를 가지며, 고정된 2바이트를 제외한 8바이트는 36^8 가지 경우의 수로 표현된다. Figure 2.1b은 패킷 분석결과에 따른 OTAC 코드생성 프로세스를 보여준다.

	Plain Text				OTAC Code			
Session	Ch1	Ch2	Ch3	Ch4	Ch1	Ch2	Ch3	Ch4
#1	1500	1600	1600	1600	A3CDCDWR	CVBRWE1T	CVBRWE1T	CVBRWE1T
#2	1500	1550	1550	1600	FEAC4FWE	FDWEGDFV	FDWEGDFV	WEFGCVBE
#3	1500	1400	1400	1600	TGW5QEYG	TH6FX7VE	TH6FX7VE	UYJGGNHG

(a) Example of captured packets



(b) Overall OTAC generation process

Figure 2.1: Packet analysis result

평문입력은 1100 ~ 1900 범위의 801가지 십진수로 표현되며, 이는 10비트 수준의 입력 다양성을 보여준다. OTAC 코드의 개별 바이트는 36가지 ASCII 코드중 하나를 사용하여 나타내며, 결과적으로 8바이트 길이의 OTAC 코드는 36^8 가지의 표현가능한 경우의 수를 가진다. 이는 42비트 수준의 출력 다양성을 보여준다.

2.1.2 엔트로피 기반 랜덤성 평가

본 연구에서는 OTAC 코드의 엔트로피에 기반한 랜덤성 평가를 수행한다. 이를 위해 개별 세션에서 생성된 64비트 길이의 OTAC 코드의 엔트로피를 측정하여 랜덤성을 분석한다. 이를 위해 425개의 OTAC 코드 변환 주기동안 동일평문에 대응되는 OTAC 코드를 수집하였다. 구체적으로, 조종기가 전송할 수 있는 전체 명령어 범위 중, 초기값에 해당하는 명령어인 1500에 대응하는 OTAC 코드를 수집하였다. Table 2.2는 본 분석에 사용한 425개의 OTAC 코드 변환 주기동안 수집된 OTAC 코드의 인덱스별 비트값을 보여준다. 엔트로피 기반 랜덤성 평가는 Approximate Entropy Test 및 Cumulative Sums (Cusum) Test로 구성된다. Table 2.1 는 엔

Table 2.1: Entropy-based randomness test result per sample

	Approximate Entropy Test	Cumulative Sums (Cusum) Test
Mean	1.00E+00	5.15E-01
TRUE	425	425
FALSE	0	0

트로피 평가결과를 보여준다. Table 2.1의 Mean은 개별 랜덤성 테스트 출력값의 평균이며, True/False는 출력값을 기반으로 판단한 OTAC 코드들의 엔트로피 기반 랜덤성 통과 여부를 보여준다.

엔트로피 기반 코드 랜덤성 평가 결과 2개의 테스트를 모두 통과하였다. 따라서 **OTAC은 엔트로피 테스트 관점에서 충분한 랜덤성을 가진것으로 분석된다.**

Table 2.2: An OTAC code sequence during 425 code update periods corresponding to the plain text '0x05DC'

Bit index	Bit sequence
0	[illegible]
1	[illegible]
2	[illegible]
3	[illegible]
4	[illegible]
5	[illegible]
6	[illegible]
7	[illegible]
8	[illegible]
9	[illegible]
10	[illegible]
11	[illegible]
12	[illegible]
13	[illegible]
14	[illegible]
15	[illegible]
16	[illegible]
17	[illegible]
18	[illegible]
19	[illegible]
20	[illegible]
21	[illegible]
22	[illegible]
23	[illegible]
24	[illegible]
25	[illegible]
26	[illegible]
27	[illegible]
28	[illegible]
29	[illegible]
30	[illegible]
31	[illegible]
32	[illegible]
33	[illegible]
34	[illegible]
35	[illegible]
36	[illegible]
37	[illegible]
38	[illegible]
39	[illegible]
40	[illegible]
41	[illegible]
42	[illegible]
43	[illegible]
44	[illegible]
45	[illegible]
46	[illegible]
47	[illegible]
48	[illegible]
49	[illegible]
50	[illegible]
51	[illegible]
52	[illegible]
53	[illegible]
54	[illegible]
55	[illegible]
56	[illegible]
57	[illegible]
58	[illegible]
59	[illegible]
60	[illegible]
61	[illegible]
62	[illegible]
63	[illegible]

2.2 코드 예측가능성 분석

2.2.1 공격자 모델

제공된 분석환경은 OTAC 코드 생성 알고리즘에 대한 정보 없이, 평문과 고정된 seed로 부터 생성되는 OTAC 코드만을 이용하여 안전성 분석을 수행한다. 평문과 OTAC 코드의 관계는 동일한 OTAC 코드 변환 주기 내에서만 유지되므로, 공격자는 동일한 OTAC 코드 변환 주기내에서 주어진 평문과 OTAC 코드쌍을 기반으로 분석을 수행한다. 이후, 공격자는 새롭게 업데이트된 OTAC 코드 변환 주기에서 수집된 OTAC 코드를 이용하여 해당 OTAC 코드 변환 주기에서 유효하게 사용가능한 OTAC 코드를 생성한다. Figure 2.2은 본 연구에서 가정한 공격 프로세스를 보여준다. **정리하면, 공격자는 주어진 OTAC 코드를 활용하여 OTAC 코드 변환 주기내에 사용가능한 새로운 OTAC 코드를 생성하는 것을 목표로한다.**

본 연구에서 가정한 공격자의 목표는 주어진 OTAC 코드 C를 이용하여, 동일한 OTAC 코드 변환 주기내에 유효한 OTAC 코드 C'을 생성하는 것이다. 유효한 OTAC 코드인 C'을 이용하여 공격자는 동일한 OTAC 코드 변환 주기동안 타겟 드론에 유효한 제어명령을 주입할 수 있다. 해당 공격을 통해 공격자는 랜덤추측 (random guessing)을 통해 OTAC 코드 공간에서 유효한 제어명령을 찾을 확률인 $\frac{2^{10}}{2^{42}} = \frac{1}{2^{32}} \approx 2.32 \times 10^{-10}$을 증가시킬 수 있다.

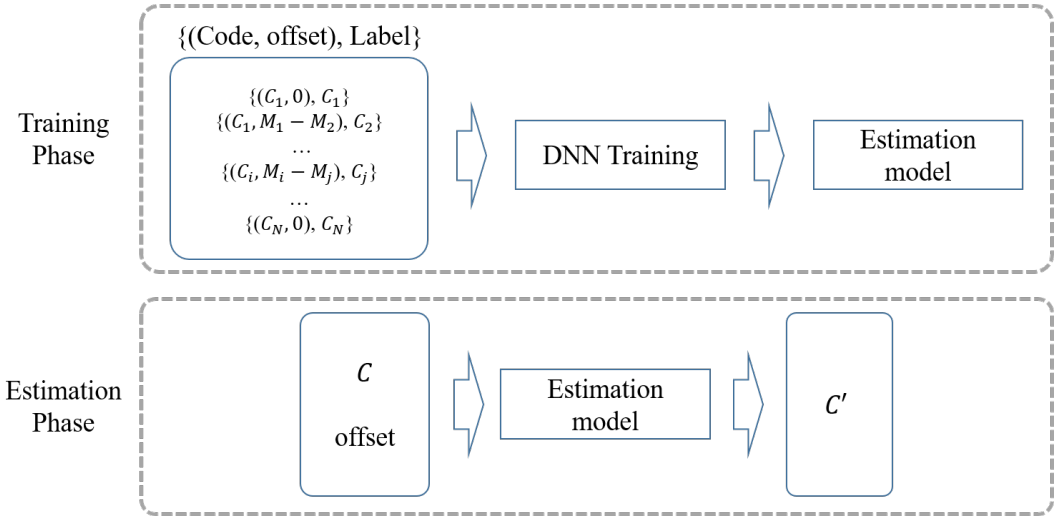


Figure 2.2: Training and estimation process

2.2.2 분석방법

OTAC 코드 변환 주기 S_i에서 평문 m_i에 대한 OTAC 코드 c_i가 주어졌을때, m'_i에 해당하는 c'_i를 생성할 경우, 공격자는 드론에게 m'_i에 해당하는 명령을 내릴 수 있다. 따라서 평문의 오프셋과 코드 변화의 관계를 학습하기 위해 m_i와 m'_i의 오프셋과 c_i를 입력으로 받아 c'_i을 출력하는 DNN 모델을 학습시킨다. 오프셋을 이용한 코드 변화 학습을 위해 다양한 평문이 필요하며 동일한 OTAC 코드 변환 주기동안 최대한 다양한 조작을 하여 데이터를 수집한다. 코드 생성 방식은 매 OTAC 코드 변환 주기마다 변경되기 때문에 오프셋은 데이터를 OTAC 코드 변환 주기별로 분할하여 OTAC 코드 변환 주기 내에서만 계산한다. 단일 OTAC 코드 변환 주기 내에는 동일한 명령어도 다수 포함되어 있으며 동일한 평문으로부터 발생하는 오프셋과 코드변화는 동일하기 때문에 중복처리를 통해 서로 다른 평문만을 사용한다. 전체 OTAC 코드 변환 주기에 대해 각 OTAC 코드 변환 주기별로 1대1 매칭을 통한 오프셋과 64비트 이진수로 변형한 전후 코드를 계산하여 학습 데이터를 수집하였으며, 총 189,596개의 학습데이터쌍을 수집하였다. Figure 2.3은 DNN 모델 학습에 사용된 코드에 해당하는 평문 제어명령어 분포 및 오프셋 분포를 보여준다.

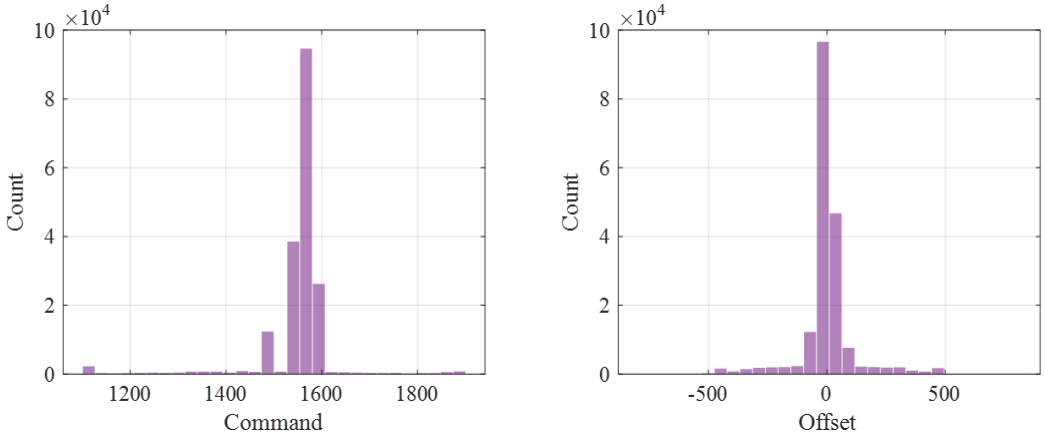


Figure 2.3: Training data distribution

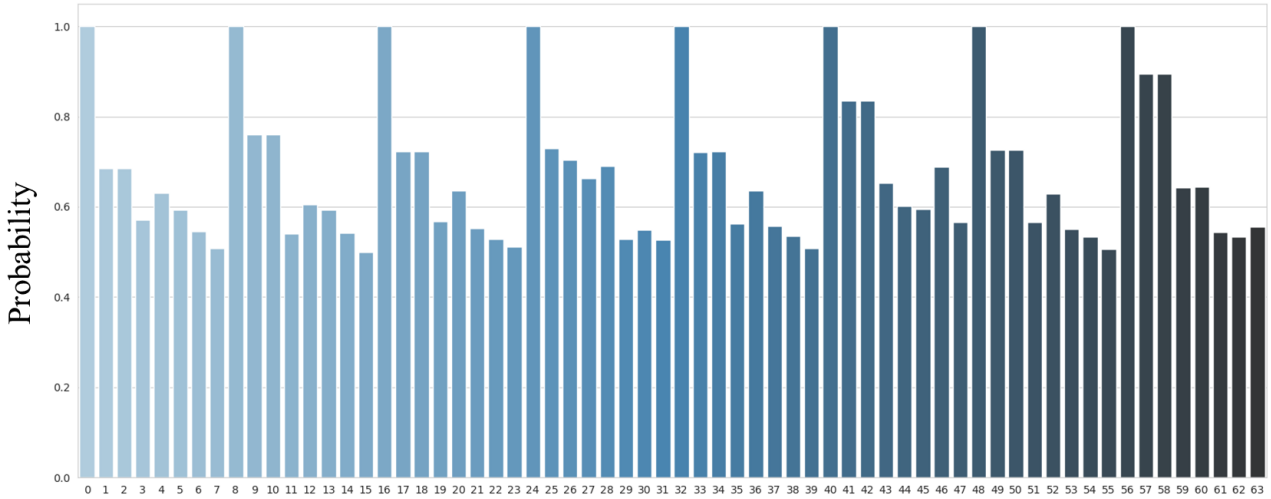


Figure 2.4: Estimation success probability according to bit position

2.2.3 분석결과

DNN 모델의 성능평가를 위해 DNN 모델의 출력값을 반올림 하여 0과 1로 구분하였으며, 개별 비트의 예측 정확도를 평가하였다. Figure 2.4은 64개의 비트별 예측 정확도를 보여준다. 각 바이트의 최상위 비트의 예측정확도가 1인 이유는 OTAC 코드의 ASCII 코드 범위가 숫자 및 대문자 알파벳으로 제한되었기 때문이다. 구체적으로, 해당 ASCII 코드의 상위 4비트는 0x3, 0x4, 0x5를 가지기 때문에 최상위 비트는 0으로 고정되어 있는 특성이 있으며, 나머지 상위 3비트 역시 엔트로피가 낮기 때문에 상대적으로 예측성공 확률이 높게 나타났다. 하지만 하위 4비트의 경우에는 예측정확도가 0.5 부근으로 형성이 되었기 때문에 랜덤 추측 수준의 정확도를 보였다. 따라서 본 연구에서 가정한 딥러닝 기반의 코드 예측 공격에 안전한 것으로 확인하였다.

> OTAC 코드 생성 알고리즘의 출력크기인 64비트 공간 중 실제로 사용되는 영역이 제한되어 있기 때문에 일부 비트의 경우 상대적으로 높은 확률로 예측에 성공하였다. 하지만, 절반 정도의 비트는 예측정확도가 0.5부근으로 형성되었기 때문에, 이는 랜덤 추측 수준의 정확도를 보이기 때문에 코드 예측 공격에 안전하다고 판단할 수 있다.

3 OTAC 적용된 drone network protocol 의 re-play attack 분석

3.1 OTAC 적용된 메시지 분석

본 연구에서 분석한 Remocopter 500 드론 시제품은 OTAC 암호문을 기반으로 조종기와 통신한다. 이를 위해 1050의 MSG ID 값을 갖는 Customized MAVLink 메세지가 사용되고 있음을 확인하였다. Customized MAVLink 메세지의 구조는 Table 3.1과 같다. ch1, ch2, ch3, ch4 필드는 조종기 조이스틱으로부터 생성된 PWM 값의 평문을 담고 있다. 그리고 rollstream, pitchstream, throttlestream, yawstream 필드는 각 채널 별 PWM 값을 OTAC 기술로 암호화한 암호문을 담고 있다. Remocopter 500 드론과 조종기에는 암호화 모드를 각각 활성화 할 수 있는 기능이 구현되어 있으며, 암호화 모드를 활성화 할 시 조종기와 드론은 암호화된 조종 명령을 기반으로 통신한다.

Table 3.1: Structure of customized OTAC MAVLink message

Field name	validotacmodecmd	otacmodecmd	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	reserved	controlcameramode	rollstream	pitchstream	throttlestream	yawstream
Byte index	0	1	2	4	6	8	10	12	14	16	18	20	22	32	42	52
Type	byte	byte	uint16	uint16	uint16	uint16	uint16	uint16	uint16	uint16	uint16	uint16	byte array	byte array	byte array	byte array
Example value (Hex)	0x01	0x02	0x044C	0x05DC	0x05DC	0x05DC	0x05DC	0x05DC	0x0100	0x0000	-	0x0001	0x3736384B343836390000	-	-	-
Example value (Integer)	1	2	1100	1500	1500	1500	1500	1500	256	0	-	1	-	-	-	-
Example value (ASCII)	-	-	-	-	-	-	-	-	-	-	-	-	768K4869	-	-	-

암호화 모드 조종기에서 암호화 모드가 활성화된 경우 드론 제어를 위한 평문 PWM 값이 OTAC로 변경되어 드론으로 송신된다. 이 때, 드론에서 암호화 모드가 활성화된 경우 수신된 데이터 스트림을 OTAC 코드에서 PWM값이 담긴 평문으로 변경한 뒤 무결성 검사를 수행한다. 무결성 검사에서 이상이 없는경우 계산된 PWM 값이 실제 flight controller로 전달된다.

3.2 OTAC 기반 MAVLink 통신 보안 평가

Customized MAVLink 메세지에 적용된 OTAC 기술의 보안성을 검증하기 위해 재전송 공격(Replay attack)을 시도하였다. 재전송 공격이란 다수의 암호문들을 미리 수집하여 복사한 후 이를 대상 노드에 재전송함으로써 암호문을 복호화 하는 노력 없이 암호 데이터의 유효성을 얻는 공격을 의미한다. 암호화 모드 활성화 전후 각각 공격을 시도하고 드론의 상태를 관찰하였다.

암호화 모드 활성화 여부와 관계 없이, 재전송 공격은 드론에 비행에 영향을 주지 못했다. 조종기는 초당 10개의 Customized MAVLink 메세지를 생성하고 10초마다 OTAC 시드가 갱신된다. 공격자는 최대 10초간 관찰한 데이터만 재전송에 활용할 수 있으므로 드론의 정상 운행에 영향을 줄 수 있을 만큼 충분한 양의 암호문을 수집할 수 없다. 따라서 OTAC은 재전송 공격으로부터 드론을 보호하는 효과적인 방어 수단이 될 수 있다.

3.3 분석 결과

공격자는 OTAC으로 보호되는 메세지를 재전송하여 드론에 영향을 줄 수 없었으며, OTAC이 조종 명령을 효과적으로 보호하고 있음을 확인했다. 하지만 Section 1.3에서 알 수 있듯이, 드론 네트워크에 침투하여 GCS 또는 조종기로 위장한 공격자는 드론을 통제할 수 있는 다양한 수단을 지니게 된다. 따라서 OTAC의 적용 범위를 드론 네트워크 통신 전반으로 확장해야한다. 모든 종류의 MAVLink 메세지에 OTAC을 적용할 경우 드론 내외부 통신의 보안성을 크게 향상시킬 수 있을 것으로 보인다.

4 결론

본 보고서는 드론과 조종기 사이의 MAVLink 프로토콜 기반 무선통신 시 OTAC 기반 암호화 통신 기술을 적용했을 때의 보안성 향상 여부를 검토하였다. MAVLink는 무인이동체를 위한 다양한 기능 구현이 포함되어있어 오픈소스 드론 뿐 아니라 다양한 상용 이동체 통신에 활용되고 있으나 한편으로는 보안 관점에서의 설계가 취약하여 메시지 위변조, 재사용과 같은 공격에 무방비로 노출되어 있다. 이러한 한계를 극복하기 위해 본 보고서의 분석 대상 드론과 조종기는 MAVLink 메시지 payload에 OTAC 기술이 적용되어 있다.

본 보고서는 OTAC 기술이 적용된 드론, 조종기, 그리고 MAVLink 메시지의 보안성을 분석하기 위해 OTAC 이 포함된 MAVLink 패킷 분석, 엔트로피 기반 랜덤성, 코드 예측 가능성, replay attack 가능성에 대한 검토를 실시하였다. 공격 모델이 OTAC이 적용되어있지 않은 MAVLink 명령 기능을 악용하고자 하는 경우 드론의 시동을 끄고 켜는 공격, Way-point를 조작하는 공격, 드론 내부에 비정상적인 부하를 발생시키는 공격 등이 가능함을 확인하였다. 그러나, OTAC에 의해 보호되는 조종기와 드론 사이의 PWM 값에 대해서는 제안된 공격 모델을 통해 메시지를 위변조하거나 재생(replay)하는 것이 불가하였다.

OTAC로 변경한 암호화코드(즉, OTAC 명령어)는 다음과 같은 보안성 향상 효과를 제공한다.

OTAC 코드의 엔트로피 기반 랜덤성 OTAC은 엔트로피 테스트 관점에서 충분한 랜덤성을 가진것으로 분석됨

OTAC 코드 예측 가능성 정상 OTAC 코드 기반 새로운 주기의 정상적인 OTAC코드 예측가능성이 낮음

Replay 공격 가능성 기전송된 OTAC 코드의 재사용되는 시기 예측이 불가능하여 Replay 공격 차단이 가능함

따라서, 기존의 MAVLink 제어명령을 OTAC 기반 암호화 코드(즉, OTAC 명령어)로 대체할 경우, 드론 하이재킹을 방지하는 보안 향상 효과를 가지게 되는 것으로 판단된다.