오픈소스 드론 취약점 분석 및 방어 기술 개발

D분과 49-Joy Security
201924523 이경민
201624587 조수현

지도 교수 손준영

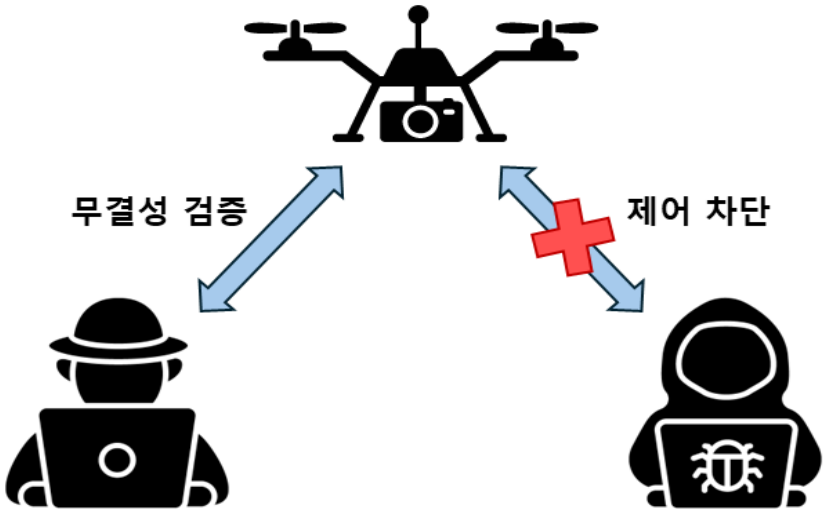
2024 전기 졸업과제

연구 배경

√ 기술 발전에 따라 드론은 많은 분야에서 적용되고, 적용하기 위해 연구됨
√ 상업용 혹은 군사적 목적의 드론은 악의적인 공격에 대해 방어할 수 있도록 지속적인 연구 필요

연구 목표

√ MAVLink 메시지에 Signature를 추가하여 무결성을 검증하고, 검증에 실패하는 경우 이후의 패킷 처리 과정을 진행하지 않도록 하여 제어를 차단하는 효과까지 얻을 수 있다.

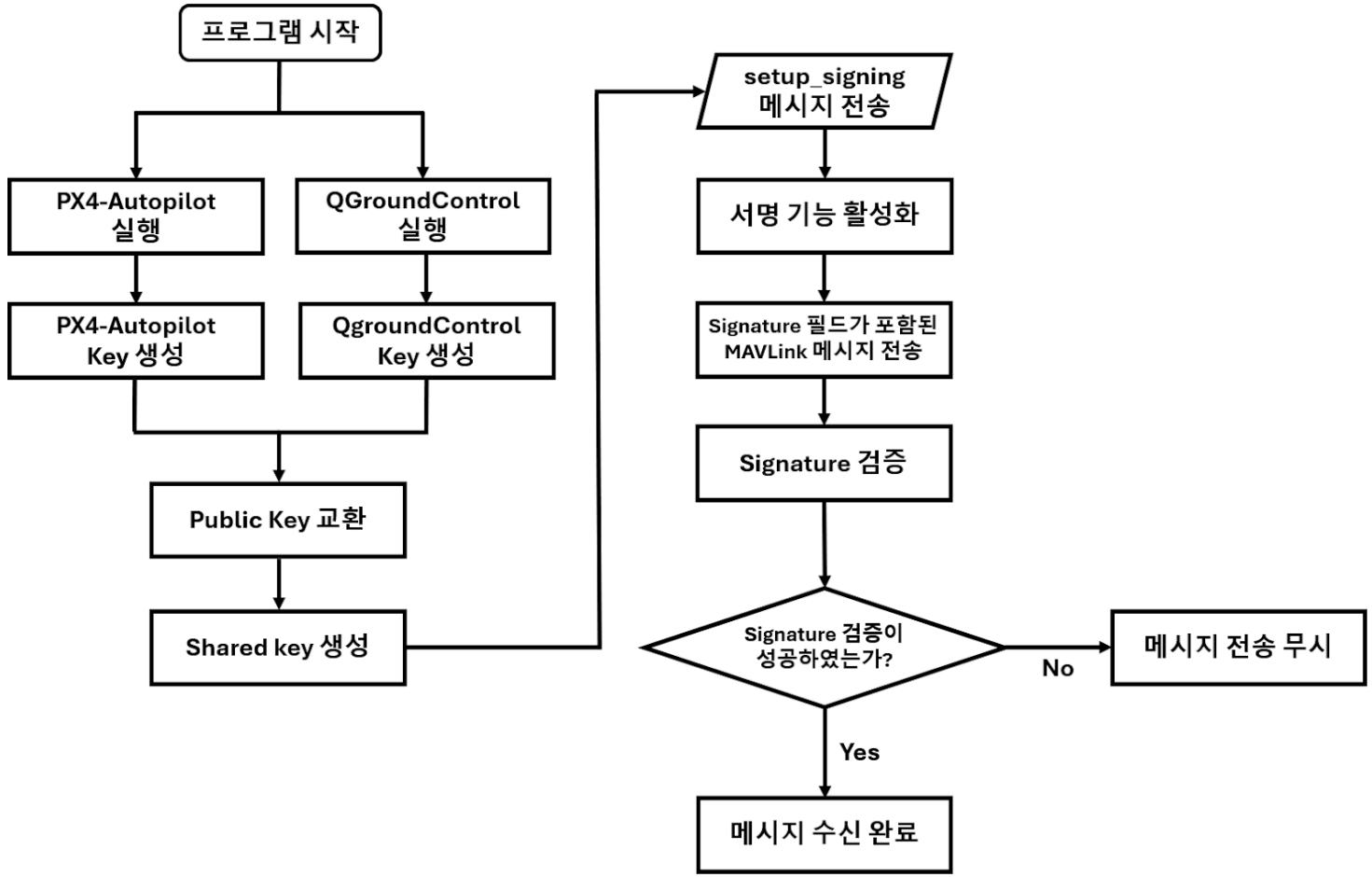


구현

√ 서명 검증을 위한 설계

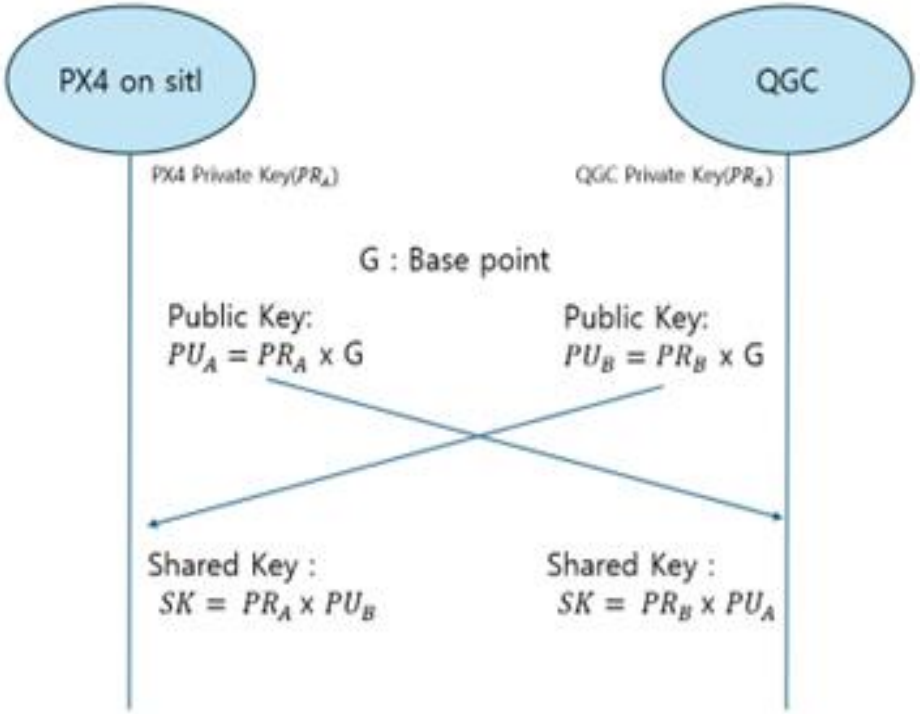
√ 서명에 필요한 Key는 ECDH 알고리즘을 통해 생성됨

√ 서명은 SHA256 알고리즘



ECDH

√ PX4와 QGC의 통신 채널에는 공개키만 노출
√ 난수로 생성된 비밀키를 기반으로 공개키 생성
√ 서로의 공개키를 교환한 뒤 자신의 비밀키와 상대방의 비밀키를 기반으로 공유 비밀키를 연산



Signature

√ SHA256 알고리즘을 적용하여 서명 생성
√ secret_key는 ECDH를 통해 생성된 공유 비밀키

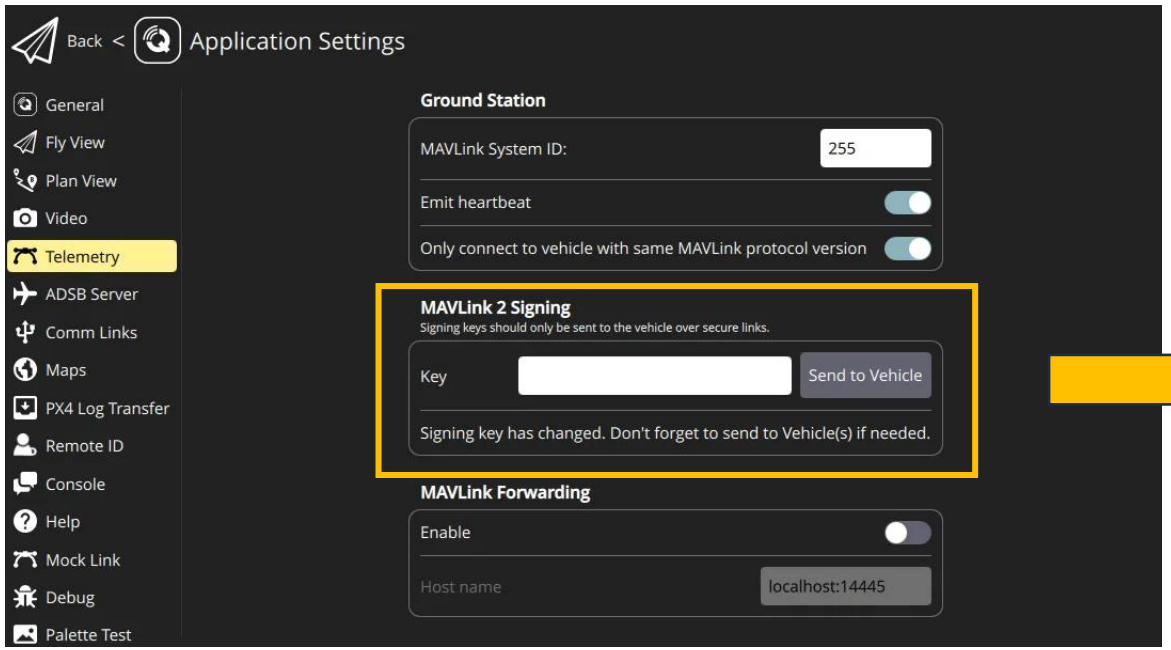
$$signature = sha256\_48(secret\_key + header + payload + CRC + link\_ID + timestamp)$$

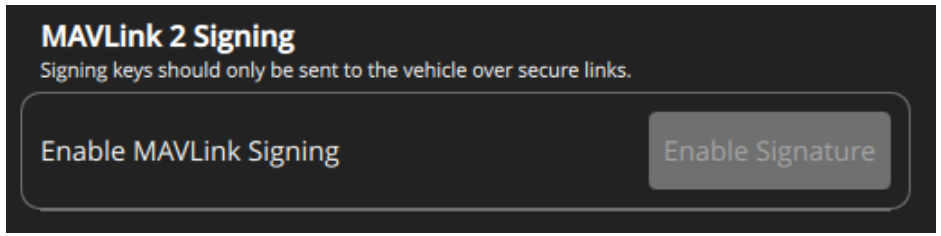
√ Signature의 구조

Link ID 1 Byte	Timestamp 6 Byte	Signature 6 Byte

Enable Signature

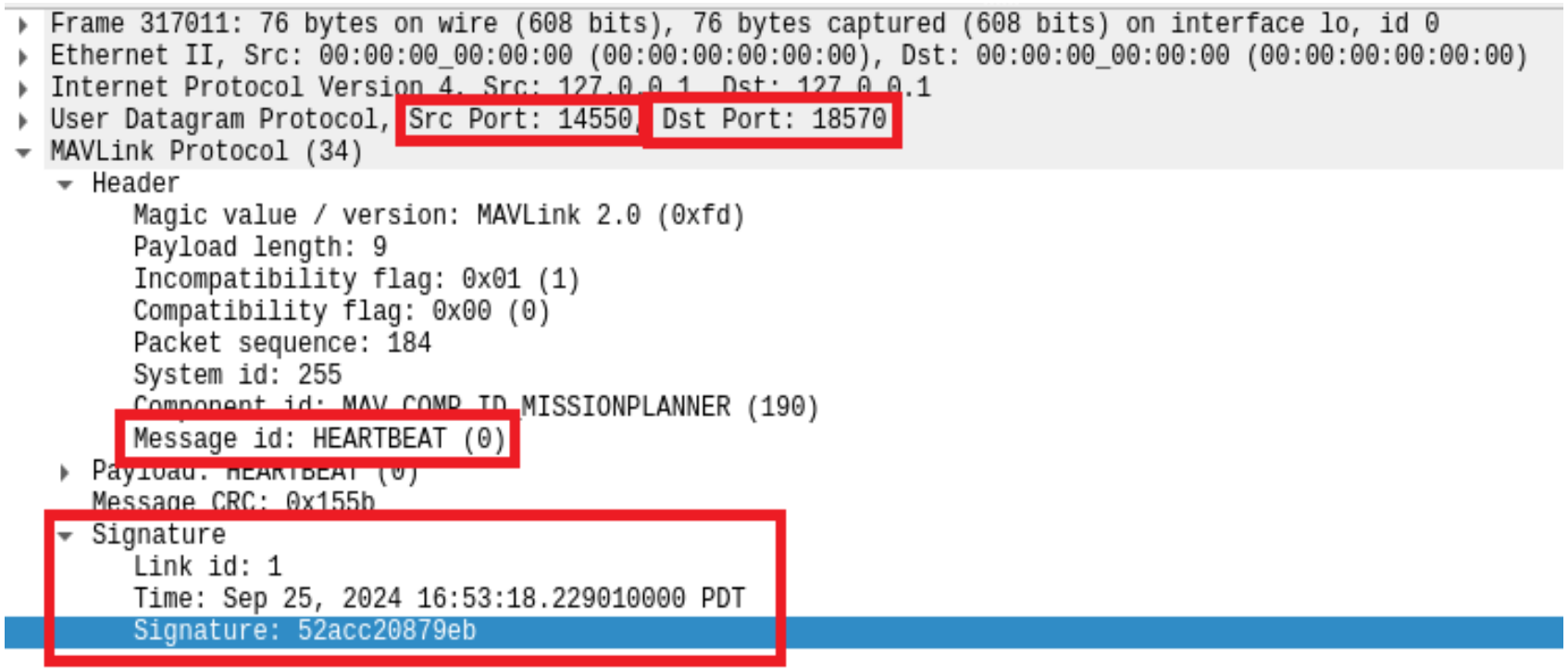
√ QGC 실행 후 Application Settings의 Telemetry 메뉴에서
"Enable Signatrue" 버튼 클릭





Enable Signature

√ 이후 전송되는 메시지에 서명이 포함된 상태로 전송됨
√ wireshark에 mavlink 플러그인을 적용하여 parsing 된 메시지 확인 가능
√ Link id는 (sys_id, component_id, link_id)의 tuple 데이터



Signature Verification

√ 오른쪽의 코드를 이용하여 전송 받은 메시지의 데이터를 하나의 배열로 조합

√ 메시지에 SHA256 해시함수를 적용해 나온 결과값을, 전송 받은 메시지의 signature와 비교하여 검증 하게 됨

```
if (message_data != nullptr) {
    uint8_t combined_data[512];
    size_t combined_length = 0;

    // shared_key
    memcpy(&combined_data[combined_length], shared_key, sizeof(shared_key));
    combined_length += sizeof(shared_key);

    // header
    combined_data[combined_length++] = msg->magic;
    combined_data[combined_length++] = msg->len;
    combined_data[combined_length++] = msg->incompat_flags;
    combined_data[combined_length++] = msg->compat_flags;
    combined_data[combined_length++] = msg->seq;
    combined_data[combined_length++] = msg->sysid;
    combined_data[combined_length++] = msg->compid;

    uint32_t msgid = msg->msgid;
    combined_data[combined_length++] = static_cast<uint8_t>(msgid & 0xFF);
    combined_data[combined_length++] = static_cast<uint8_t>((msgid >> 8) & 0xFF);
    combined_data[combined_length++] = static_cast<uint8_t>((msgid >> 16) & 0xFF);

    // payload
    memcpy(&combined_data[combined_length], message_data, message_length);
    combined_length += message_length;

    // CRC
    uint16_t crc = msg->checksum;
    combined_data[combined_length++] = static_cast<uint8_t>(crc & 0xFF);
    combined_data[combined_length++] = static_cast<uint8_t>((crc >> 8) & 0xFF);

    // link-ID
    combined_data[combined_length++] = msg->signature[0];

    // timestamp
    memcpy(&combined_data[combined_length], &msg->signature[1], 6);
    combined_length += 6;
```

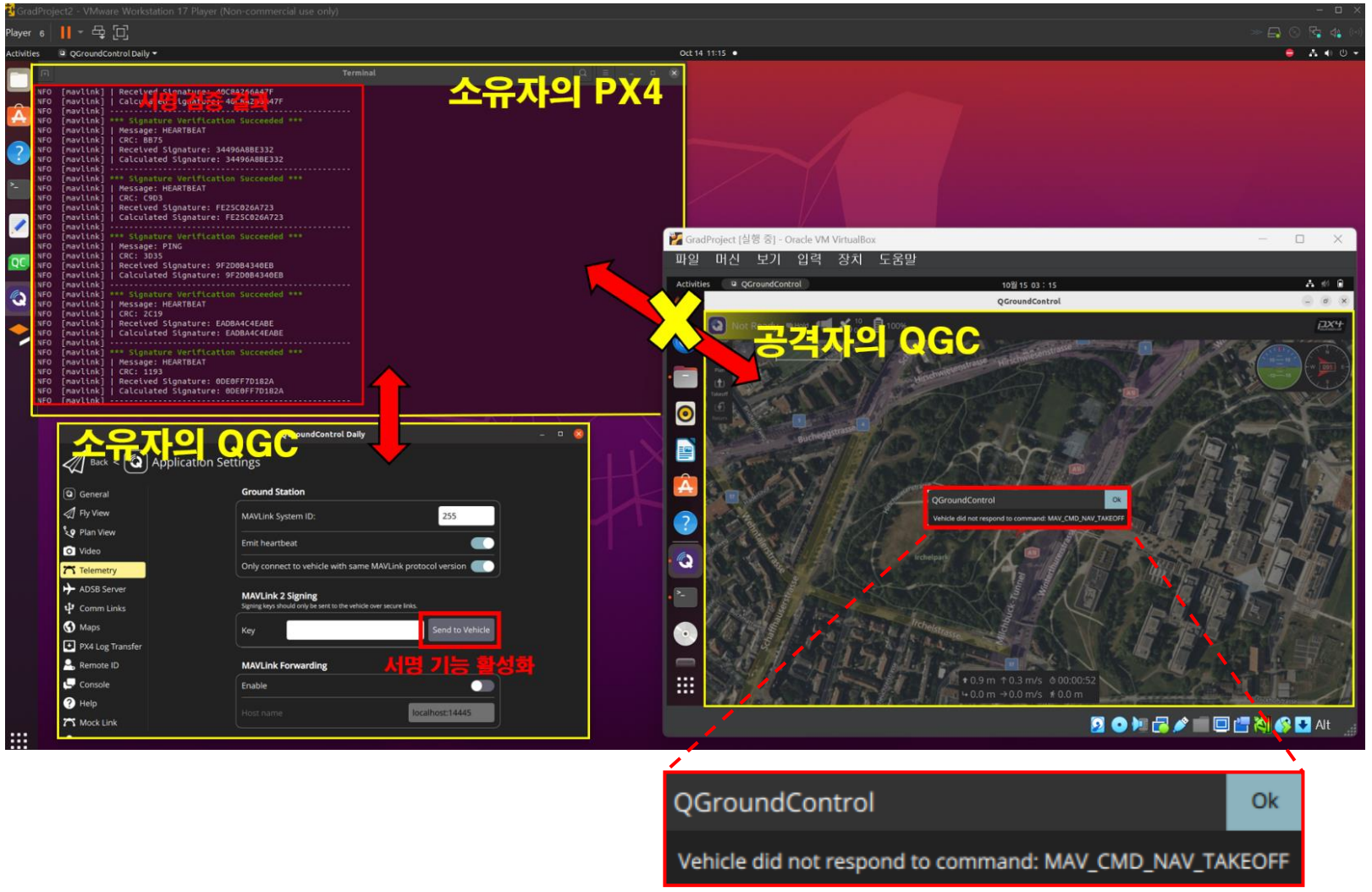
Signature Verification


```
INFO  [mavlink] *** Signature Verification Succeeded ***
INFO  [mavlink] | Message: PING
INFO  [mavlink] | CRC: 46E7
INFO  [mavlink] | Received Signature: B8A41BA73CB5
INFO  [mavlink] | Calculated Signature: B8A41BA73CB5
INFO  [mavlink] ------------------------------------------------
INFO  [mavlink] *** Signature Verification Succeeded ***
INFO  [mavlink] | Message: HEARTBEAT
INFO  [mavlink] | CRC: C9D3
INFO  [mavlink] | Received Signature: 831E2CEB90BB
INFO  [mavlink] | Calculated Signature: 831E2CEB90BB
INFO  [mavlink] ------------------------------------------------
INFO  [mavlink] *** Signature Verification Succeeded ***
INFO  [mavlink] | Message: COMMAND_LONG
INFO  [mavlink] | CRC: 0C05
INFO  [mavlink] | Received Signature: 1B8E0CD26298
INFO  [mavlink] | Calculated Signature: 1B8E0CD26298
INFO  [mavlink] ------------------------------------------------
INFO  [mavlink] *** Signature Verification Succeeded ***
INFO  [mavlink] | Message: COMMAND_LONG
INFO  [mavlink] | CRC: D4D3
INFO  [mavlink] | Received Signature: 17B0BC4673B1
INFO  [mavlink] | Calculated Signature: 17B0BC4673B1
INFO  [mavlink] ------------------------------------------------
INFO  [commander] Armed by external command
INFO  [tone_alarm] arming warning
INFO  [mavlink] *** Signature Verification Succeeded ***
INFO  [mavlink] | Message: HEARTBEAT
INFO  [mavlink] | CRC: 2C19
INFO  [mavlink] | Received Signature: A75A6C36E28C
INFO  [mavlink] | Calculated Signature: A75A6C36E28C
INFO  [mavlink] ------------------------------------------------
INFO  [commander] Takeoff detected
INFO  [mavlink] *** Signature Verification Succeeded ***
INFO  [mavlink] | Message: SET_MODE
INFO  [mavlink] | CRC: BC07
INFO  [mavlink] | Received Signature: 06469D567FAC
INFO  [mavlink] | Calculated Signature: 06469D567FAC
INFO  [mavlink] ------------------------------------------------
```

```
INFO  [mavlink] *** Signature Verification Failed ***
INFO  [mavlink] | Message: COMMAND_LONG
INFO  [mavlink] | CRC: 1C83
INFO  [mavlink] | Received Signature: F92865DF5156
INFO  [mavlink] | Calculated Signature: B13E36ADD3E7
INFO  [mavlink] ------------------------------------------------
INFO  [mavlink] *** Signature Verification Failed ***
INFO  [mavlink] | Message: HEARTBEAT
INFO  [mavlink] | CRC: A2CF
INFO  [mavlink] | Received Signature: 8C0962DA3DDD
INFO  [mavlink] | Calculated Signature: 606E59F73F65
INFO  [mavlink] ------------------------------------------------
INFO  [mavlink] *** Signature Verification Failed ***
INFO  [mavlink] | Message: HEARTBEAT
INFO  [mavlink] | CRC: AD8C
INFO  [mavlink] | Received Signature: B0A8072E30AE
INFO  [mavlink] | Calculated Signature: F418E9685C95
INFO  [mavlink] ------------------------------------------------
INFO  [mavlink] *** Signature Verification Failed ***
INFO  [mavlink] | Message: COMMAND_LONG
INFO  [mavlink] | CRC: 0400
INFO  [mavlink] | Received Signature: 2F83BED83173
INFO  [mavlink] | Calculated Signature: 340C1B484CDB
INFO  [mavlink] ------------------------------------------------
INFO  [mavlink] *** Signature Verification Failed ***
INFO  [mavlink] | Message: HEARTBEAT
INFO  [mavlink] | CRC: DF2A
INFO  [mavlink] | Received Signature: 652D239A9093
INFO  [mavlink] | Calculated Signature: 3592687C1848
INFO  [mavlink] ------------------------------------------------
```

√ 서명 검증에 성공 했을 때와 실패 했을 때의 출력 화면

Signature Verification

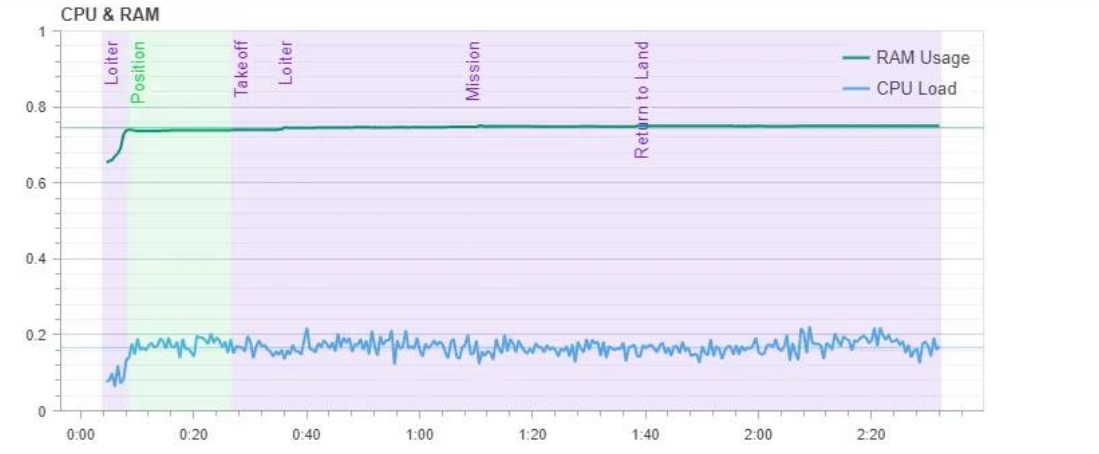


√ 공격자가 PX4의 UDP port로 접속을 하여 연결에 성공하여도, 서명이 활성화 되어 있는 상태라면, 공격자의 제어 명령이 무시된다.

√ 키 값이 노출되어 공격자가 서명을 생성할 수 있게 되어도, 서명 스트림이 공격자의 system id와 component id를 가리키지 않기 때문에 서명 검증에 실패하게 된다.

연구 결과 – CPU & RAM 사용량

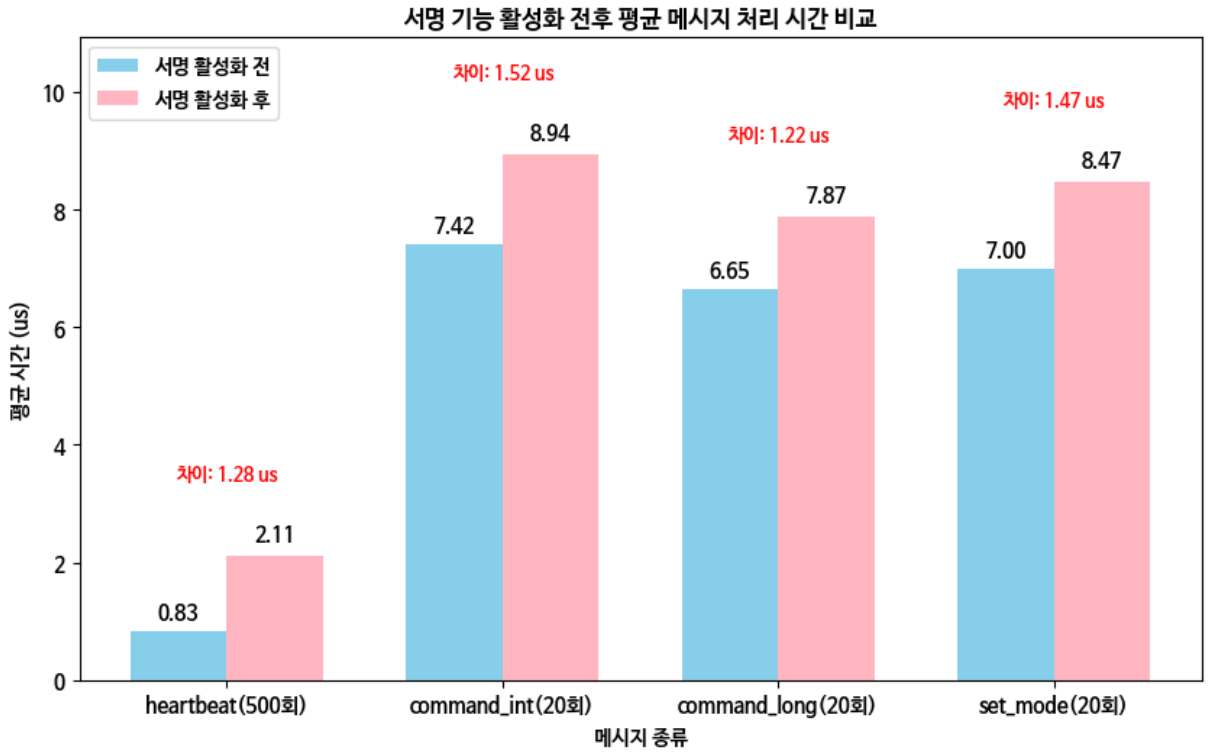
√ 서명 활성화 이전
√ 평균 CPU 사용량 약 19%
√ 평균 RAM 사용량 약 73%



√ 서명 활성화 이후
√ 평균 CPU 사용량 약 19%
√ 평균 RAM 사용량 약 74%

√ 두 개의 지표에서 거의 동일한 사용량
√ 유의미한 차이 X

연구 결과 – 메시지 처리 시간



√ 드론 제어와 연결 관리에 사용되는 메시지에 대하여 메시지 처리 시간 측정

√ 서명 활성화 전후 메시지에 대한 처리 시간을 측정하여 평균 시간 계산

√ 대략 1.2 μs ~ 1.5 μs 시간 차이 발생

√ 해당 시간은 서명 검증에 필요한 시간

감사합니다.