*Journal of The Korea Institute of Information Security & Cryptology* ISSN 1598-3986(Print)
VOL.31, NO.2, Apr. 2021 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2021.31.2.133

# 양자 엔트로피 기반 난수 발생기를 이용한 드론 제어 데이터 보안 연구*

김 태 완,[1†] 이 세 윤,[1] 정 서 우,[1] 위 한 샘,[1] 이 옥 연[2‡]
[1,2]국민대학교 (대학원생, 교수)

# A Research on the Security of Drone Control Data Using Quantum Entropy-Based Random Number Generator*

Tae-Wan Kim,[1†] Se-Yoon Lee,[1] Seo-Woo Jung,[1]
Han-saem Wi,[1] Ok-yeon Yi[2‡]
[1,2]Kookmin University (Graduate student, Professor)

## 요 약

Ardupilot, PX4는 드론에 탑재되는 대표적인 오픈소스 프로젝트이며 개발비용 절감, 소스코드의 안정성, 신속한 개발 피드백 등의 이유로 드론 제어 오픈소스들은 국내·외에서 드론 개발에 널리 사용되고 있다. 하지만 이와 같은 장점 이면에는 치명적인 단점 또한 존재한다. 여러 기관에서 사용되고 있는 드론에 탑재된 오픈소스들이 데이터 보안에 대한 취약점이 존재할 경우, 해당 기관에서 사용되는 드론은 모두 잠재적으로 보안 공격의 대상이 되며 이는 환경의 특성에 따라 연쇄적인 경제적, 인명적 손실을 불러올 수 있다.

본 논문에서는 Ardupilot, PX4에서 사용하고 있는 MAVLink v1.0, MAVLink v2.0의 데이터 보안 취약점을 설명하고 이에 대응할 수 있는 양자 기반 난수 발생기를 활용한 개체 인증, 데이터 보안을 제안하며 제안된 데이터 보안의 성능 측정을 통해 가용성 분석을 수행한다.

## ABSTRACT

Ardupilot and PX4 are representative open source projects for drones, and drone-controlled open sources are widely used for drone development at home and abroad due to reduced development costs, stability of source code, and rapid development feedback. However there are also fatal drawbacks behind such advantages. Open-source on drones used by various agencies If there are vulnerabilities in data security, all drones used by those agencies are potentially subject to security attacks, this could lead to a series of economic and human losses, depending on the nature of the environment.

In this paper, we explain the data security vulnerabilities of MAVLink v1.0, MAVLink v2.0 used in Ardupilot, PX4, and propose object authentication using quantum entropy-based random number generator to respond to them, propose data security, and conduct availability analysis through performance measurement of proposed data security.

**Keywords:** UAS, Drone, MAVLink, Quantum random number, IoT

Received(10. 16. 2020), Modified(02. 22. 2021), Accepted(02. 22. 2021)

* 본 연구는 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No. 2020-0-00085, 5G+ 기반 6G 이동통신 정보보안 기술 연구)

* 본 논문은 2020년도 한국정보보호학회 하계학술대회에서 발표한 우수논문을 개선 및 확장한 것임

† 주저자, ktw0308@kookmin.ac.kr

‡ 교신저자, oyyi@kookmin.ac.kr(Corresponding author)

# I. 서 론

드론은 군사 분야용, 산업 분야, 민간 분야 등 여러 분야에 광범위하게 활용되면서 항공, ICT, SW, 센서 등의 기술들과 융합되어 4차 산업혁명에서 핵심적인 역할을 수행한다[1]. 이처럼 많은 산업과의 융합을 통해 드론에 유입되는 정보의 종류가 다양해지고 있으며, 드론을 통해 취득할 수 있는 정보의 중요도가 높아졌다. 따라서, 상응하는 드론 보안 기술에 대한 요구 또한 증가하였다.

일반적으로, 임무 수행 기반의 드론 시스템은 지상조종국과 드론 간의 전용 메시징 프로토콜을 기반으로 한 통신으로 운영된다. 만약 지상조종국과 드론 간의 메시징 프로토콜에 보안 취약점이 존재하거나 드론에 보안이 설계되어 있지 않다면, 그를 이용한 데이터 유출, 데이터 위·변조 등의 보안 위협이 발생할 수 있고, 이는 보안 사고로 이어져 막대한 경제적 피해와 인명 피해가 발생할 수 있다. 2016년 일본 도쿄에서 열린 보안 콘퍼런스 '2016 PacSec'에서는 DSMx 프로토콜의 보안 취약점을 이용하여 레저용 드론들을 해킹해 조종권을 빼앗는 상황을 시연하였다. 많은 드론에서 사용하고 있는 메시징 프로토콜인 MAVLink(Micro Air Vehicle communication protocol) 또한 보안 취약점이 존재하기 때문에 하이재킹, 도청 등의 공격이 가능하다[2]. 이러한 보안 취약점을 보완하기 위해 드론 기반 서비스 보안요구사항에서는 지상조종국과 드론 간의 메시지에 대한 기밀성 제공과 개체 인증이 필요하다[3]. 기밀성은 암호 알고리즘을 이용하여 제공할 수 있다. 특히, 블록암호 알고리즘에서의 안전성은 Kerckhoff의 원칙에 따라 암호키의 안전성에 기반하며, 해당 암호키는 개체 간 키 교환 또는 키 공유 과정을 통해 생성된다. 이처럼 인증 및 암호화에 사용되는 대부분의 키 교환 과정에서는 난수가 요구된다. 그러나 현재 운용되는 많은 드론은 내부 운영시스템에서의 난수 생성을 위한 충분한 잡음원 수집이 어렵기 때문에 난수를 이용한 인증 및 암호화 기술을 도입하기 힘들다.

이에 따라 본 논문에서는 드론 내에 양자 엔트로피 기반 난수 발생기를 탑재하여 암호학적으로 안전한 난수를 드론의 개체 인증 및 키 교환에 적용한다. 또한 키 교환 결과를 바탕으로 대칭키 암호 알고리즘을 이용하여 데이터의 기밀성을 제공하고, MAVLink의 보안 취약점 분석 결과를 이용하여 보안 기능 적용 시 통신에서 발생하는 시간에 따른 성능 측정 결과를 제시한다.

# II. MAVLink 취약점 분석

## 2.1 UAS 통신 구조

UAS(무인항공 시스템)는 드론(무인항공기)과 지상조종국(지상제어 시스템)로 구성되어 있다[4]. Fig. 1.은 UAS의 구성도를 나타낸다.

드론은 무인으로 미션을 수행하는 비행물체이다. 드론의 작동을 위한 소프트웨어 플랫폼으로는 ArduPilot, Paparazzi, PX4, MultiWii 등이 있으며, 본 논문에서 사용할 소프트웨어 플랫폼은 PX4이다. PX4는 드론 개발자들이 기술을 공유해 드론 애플리케이션에 맞는 맞춤형 솔루션을 제공한다. 또한, 드론 하드웨어 지원과 소프트웨어 스택 표준을 제공하여 확장 가능한 방식으로 하드웨어와 소프트웨어를 구축하고 유지·관리할 수 있다[5].

지상조종국은 드론과 통신하여 제어 정보를 전달하고, 드론의 상태를 확인하는 프로그램 및 장치이다. 지상조종국과 드론은 WiFi, 블루투스, 이동통신 등 다양한 방법으로 통신하며, 통신 프로토콜로는 MultiWii, MAVLink 등이 있다.

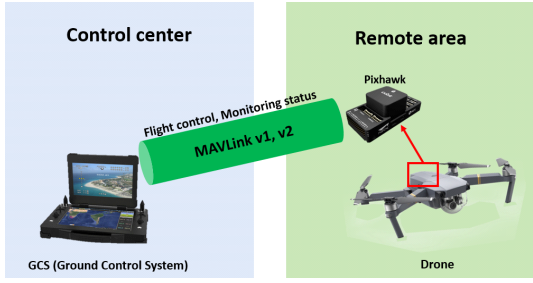


Fig. 1. UAS construction

## 2.2 MAVLink

MAVLink는 소형 무인 장치의 비행 스택과 지상조종국 간의 통신을 위해 만들어진 애플리케이션 계층에서의 경량 메시징 프로토콜이다.

MAVLink는 파워 부족과 버퍼의 제약 등 컴퓨터 자원 부족 문제로 통신을 위한 연산은 최소화하여 패킷 구조가 단순하다. Lorenz Meier에 의해 2009년 초에 처음 발표된 MAVLink는 현재 v2까지 나왔고, 오픈소스를 통해 활발하게 개발 프로젝트가 진

행 중이다[6].

Fig. 2.는 MAVLink v1 메시지 형식을 나타낸다. Table 1.은 Fig. 2.의 각 필드에 대한 설명을 나타낸다.

MAVLink v1은 메시지 타입 표현이 제한적이고, 보안 측면에서의 인증 및 무결성 서비스를 제공하지 않는다. 이를 보완하기 위해 MAVLink v2는 MAVLink v1에 확장성과 보안이 추가되었다.

Fig. 3.은 MAVLink v2 메시지 형식을 나타내며, Table 2.는 MAVLink v1에서 추가된 필드에 대한 설명을 나타낸다.

MAVLink v2의 MSG ID 필드는 확장성을 위해 기존의 MSG ID 필드보다 2 bytes가 추가되었다. 대표적으로 사용하는 MSG ID로는 Command ID에 따라 필요한 명령을 전달하는 0x4c (COMMAND_LONG), 명령에 따른 ACK

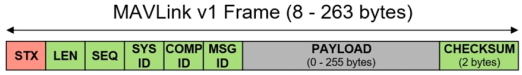


Fig. 2. MAVLink v1 frame[6]

Table 1. Description of MAVLink.v1 field[6]

| Field (Size) | Description |
|---|---|
| STX (1 byte) | - Specify version<br>- MAVLink v1 : 0xfe |
| LEN (1 byte) | - Length of MAVLink protocol's payload |
| SEQ (1 byte) | - Sequence of MAVLink packet<br>- Range : 0~255<br>- Initialize to 0 when exceeded 255<br>- Can check packet loss status |
| SYS ID (1 byte) | - System ID<br>- FC(Flight Controller) : 0~254<br>- GCS : Fixed at 255 |
| COM ID (1 byte) | - ID by component |
| MSG ID (1 byte) | - Type of message |
| PAYLOAD (0~255 bytes) | - Data of MAVLink application |
| CHECKSUM (2 bytes) | - Header values excluding STX<br>- CRC-16 values of Payload<br>- Provide message integrity |

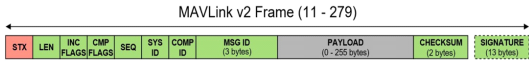


Fig. 3. MAVLink v2 frame[6]

Table 2. Description of MAVLink.v2 field[6]

| Field (Size) | | Description |
|---|---|---|
| STX (1 byte) | | - Specify version<br>- MAVLink v2 : 0xfd |
| INC FLAGS (1 byte) | | - Whether to use signature<br>- 0x01 : Use signature |
| CMP FLAGS (1 byte) | | - Priority of MAVLink protocol packets |
| MSG ID (3 bytes) | | - Type of message |
| SIGNATURE | link id (1 byte) | - Link ID to which packets are sent<br>- Used to distinguish between multiple channels |
| | timestamp (6 bytes) | - GMT time as of January 1, 2015<br>- Unit: Microseconds |
| | signature (6 bytes) | - Using SHA256<br>- Input: Message, timestamp, secret key<br>- Output: msb 48-bit of hash result |

메시지를 전송하는 0x4d (COMMAND_ACK)가 있다. Fig. 4.는 MSG ID가 0x4c와 0x4d일 경우의 패킷 교환을 나타낸다.

또한 MAVLink v2는 보안성을 위해 패킷의 끝에 signature 필드를 추가하여, 전송되는 데이터에 대한 해시함수를 통한 무결성과 비밀키(공유키)를 이용해 출처 인증을 제공한다.

$$SHA\,256(\text{메시지} \| time\,stamp \| \text{비밀키}) \qquad (1)$$

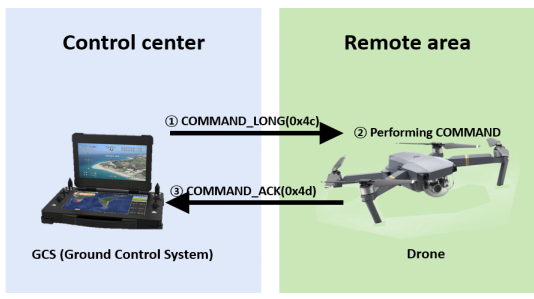


Fig. 4. Example of MAVLink v2 packet exchange

(1)의 결과로 나온 값의 최상위 비트 48-bit를 signature로 사용한다.

## 2.3 MAVLink v2 프로토콜 보안 취약점

### 2.3.1 기밀성 제공의 부재

기밀성은 허락되지 않은 사용자가 정보의 내용을 알 수 없도록 하는 것이다. MAVLink v1 및 v2는 연산을 최소화하여 암호화를 하지 않아 기밀성을 제공하지 않는다. Table 3.은 이를 입증하기 위한 패킷 캡처 환경을 나타낸다. Fig. 5.는 Table 3.의 환경에서의 전체적인 구성도를 나타낸다.

Ubuntu 18.04 운영체제에서 지상조종국은 QGC(QGroundControl), 드론은 GAZEBO로 모의 환경을 구축하였다. QGC와 GAZEBO는 UDP를 이용하여 통신하며, Wireshark로 MAVLink v2 패킷을 캡처하였다.

Fig. 6.에 표시된 패킷은 MAVLink v2의 구조를 따른다. 패킷에서 MAVLink 버전(0xfd), MSG ID(0x4c) 등의 헤더 정보가 원본으로 송신되는 것을 확인할 수 있다.

Table 4.는 MSG ID에 따른 명령을 포함한 payload를 나타낸다. 이륙 명령을 나타내는 command 필드 값(0x16)과 드론의 위치, 시스템

Table 3. Description of MAVLink.v2 field

| | GCS | Drone |
|---|---|---|
| OS | Ubuntu 18.04 | |
| Application | QGC | GAZEBO |
| Packet sniffer | Wireshark | |

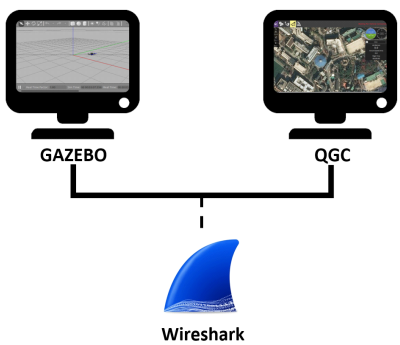


Fig. 5. Virtual environment diagram

```
e0 d5 5e 9f bb c5 b4 2e  99 28 4e f0 08 00 45 00
00 48 4f fb 40 00 40 11  e8 9b c0 a8 40 2f c0 a8
40 8e 38 f4 38 f4 00 34  b6 04 fd 20 00 00 de ff
be 4c 00 00 00 00 80 bf  00 00 00 00 00 00 00 00
00 00 c0 7f 00 00 c0 7f  00 00 c0 7f 0c 02 23 40
16 00 01 01 32 9b
```

Fig. 6. Example of packet capture

Table 4. Payload by MSG ID(0x4c)[6]

| Field | Data type | Description |
|---|---|---|
| param1 | float | Minimum pitch |
| param2 | float | Empty |
| param3 | float | Empty |
| param4 | float | Yaw angle |
| param5 | float | Latitude |
| param6 | float | Longitude |
| param7 | float | Altitude |
| command | uint16_t | Value: 0x16 (take-off) |
| target_ system | uint8_t | System on which to execute the command |
| target_ componet | uint8_t | Components to execute commands |
| confirmation | uint8_t | Confirmation |

및 컴포넌트에 대한 데이터가 원본 데이터로 송신되는 것을 확인할 수 있다. 따라서 MAVLink v2는 기밀성을 제공하지 않는다.

### 2.3.2 안전하지 않은 무결성 및 출처 인증

무결성은 비허가된 사용자가 정보를 수정할 수 없도록 하는 것이다. MAVLink 프로토콜 checksum 필드는 CRC를 사용하는 필드로서, 위·변조 검출이 아닌 네트워크 상에서의 손실 등의 단순 오류 검출만을 위해 사용된다.

또한 MAVLink v2에서의 SIGNATURE-signature 필드는 해시값의 msb 48-bit만을 사용한다. 이는 생일 역설에 따르면 약 $\frac{1}{2^{24}}$의 확률로 충돌쌍을 발견할 수 있다. 기존의 SHA-256의 충돌쌍 발견 확률인 $\frac{1}{2^{128}}$보다 큰 값을 가지므로, 제 2 역상 저항성이 낮아진다[7].

$$SHA256(M \| KEY) = SHA256(M' \| KEY) \quad (2)$$

$$SHA256(M \| KEY) = SHA256(M \| KEY') \quad (3)$$

(2)의 경우 원본 데이터와 동일한 해시값을 가지는 $M$과 다른 $M'$을 찾을 수 있다. 따라서 완전한 무결성을 보증하지 못한다. (3)의 경우 원본 데이터와 동일한 해시값을 가지는 $KEY$와 다른 $KEY'$를 찾을 수 있다. 따라서 인증되지 않은 사용자로부터 접근이 가능하다.

(2), (3)에 의해 MAVLink v2의 무결성 알고리즘 및 인증 메커니즘은 안전하지 않다.

MAVLink v2의 기밀성 제공의 부재와 안전하지 않은 무결성 및 인증을 확인하였다. 이에 따라 본 논문에서는 난수 생성을 통한 개체 인증 및 암호 키 교환을 수행하고, 블록암호 알고리즘을 통한 기밀성 제공을 수행한다.

## III. 보안 연구

### 3.1 드론의 난수 생성 제약

드론 통신은 도청 및 데이터 위·변조 공격과 드론의 GPS 스푸핑 공격 등에 취약하다[8], [9]. 이와 같은 공격으로부터 드론을 보호하기 위해서 많은 보안 시스템이 존재하지만, 자원과 전력의 제약이 있는 시스템에서는 최적이 아니기 때문에 적용되지 못하고 있다[10].

기존의 다양한 암호 시스템에서는 가장 대중적으로 사용되고 있는 암호 라이브러리인 OpenSSL이나 다양한 방법으로 기기의 내·외부의 자원을 이용하여 난수를 생성한다. 또한 드론 운영체제와는 다르게 윈도즈, 리눅스 운영체제에서는 디스크 읽기·쓰기 시간 및 인터럽트 요청 시간 등의 잡음원을 제공하는 함수가 존재한다[11]. 하지만 드론은 이러한 충분한 잡음원을 사용할 수가 없기 때문에 좋은 비트를 갖는 난수를 생성하기 어렵다[12]. Table 5.는 드론에서는 제공하지 않는 Windows, Linux 운영체제의 잡음원 제공 함수이다.

지상조종국과 드론 간의 인증 및 키 교환 과정에서는 난수가 필요하며, 난수의 안전성은 드론의 안전성으로 직결된다[8]. 그러나 드론은 데스크톱 PC처럼 마우스나 키보드와 같은 외부장치나, 내부 시간

Table 5. Noise source function of Windows and Linux[11]

| OS | Noise source function of OS |
|---|---|
| Windows | Windows random number generation |
| | Remaining disk space |
| | System interrupt information |
| | Position of mouse cursor |
| Linux | Linux random number generation |
| | Size of the disk used |
| | Information of ethernet |
| | Interrupts(IRQ) information generated by device |

등 충분한 자원을 seed로 사용할 수 없기 때문에 난수성을 보장할 수 없다[12]. 또한, 충분한 자원을 seed로 사용할 수 있다고 하더라도 드론 내부에 별도의 DRBG가 없기 때문에 드론에서는 안전한 난수를 생성해 낼 수 없다.

### 3.2 양자 엔트로피 기반 난수 발생기

난수는 암호화 시스템을 구성하는데 필수적인 요소로, 예측 불가능성, 비편향성, 독립성을 갖추고 있으면 암호학적으로 안전한 난수로 간주된다.

Fig. 7.은 암호학적으로 안전한 난수 발생기 구조를 나타낸다. 암호학적으로 안전한 난수 발생기는 TRNG(진난수 발생기)와 PRNG(의사 난수 생성기)가 결합된 구조이다. 이는 PC의 잡음원 등을 TRNG의 입력값으로 하여 나온 출력값을 PRNG의

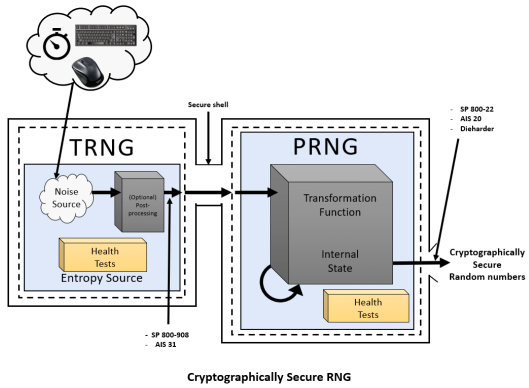


Fig. 7. Cryptographically secure random number generator structure

seed로 사용하여 암호학적으로 안전한 난수를 생성하는 구조이다. 그러나 드론에서는 충분한 잡음원을 추출하기 어렵기 때문에 이러한 난수 발생기의 구조를 사용하기 어렵다.

이를 해결하기 위한 방법으로는 QRNG(양자 엔트로피 기반 난수 발생기)의 활용이 있다[13]. 최근 IoT 보안 위협이 증가하면서 양자 난수를 활용한 암호화 기술들이 개발 중이다. 양자 난수는 양자역학 현상을 활용하여 기존의 난수 발생기인 TRNG와 PRNG 출력값의 문제점을 보완한 완벽한 난수로 평가받는다.

양자 난수를 추출할 수 있는 방법으로는 빛의 무작위성을 이용하는 방법과 방사성 동위원소의 붕괴를 이용하는 방법이 있다. QRNG는 양자 현상으로부터 난수를 생성하는 장치이므로, 양자물리학의 원리에 따라 어떤 방법으로도 예측할 수 없다[13]. 따라서 양자 난수를 생성하는 Micro-QRNG를 사용하여 안전한 보안 시스템을 구현한다면 드론에서의 난수 발생기의 문제를 해결할 수 있다.

### 3.3 개체 인증

개체 인증이란 권한이 없는 사용자가 중요한 정보에 접근하지 못하게 하도록 개체 간 상대의 신분을 검증하는 것이다. DH나 ECDH 같은 키 교환 메커니즘은 중간자 공격에 취약하다. 이를 방지하기 위해서는 키 교환을 수행하기 전에 개체 인증을 수행해야 한다[14]. 개체 인증에는 패스워드와 같은 약한 개체 인증과 Challenge-Response 방식, Zero-Knowledge 프로토콜 등 강한 개체 인증이 있다. 강한 개체 인증 중 널리 사용되고 있는 Challenge-Response 인증 방식은 검증자가 증명자의 신분을 확인하는 단방향 인증과 증명자와 검증자 간의 신분을 상호 확인하는 양방향 인증이 있다.

개체 인증은 난수와 암호 알고리즘을 이용한다. 난수와 대칭키 암호 알고리즘을 사용하는 양방향 Challenge-Response 방식 중 CHAP(Challenge Handshake Authentication Protocol) 인증 과정은 Fig. 8.과 같다[15].

1. 증명자와 검증자는 각각의 난수 $r_p$, $r_v$ 생성
2. 증명자는 검증자에게 난수 $r_p$ 전송
3. 검증자는 생성한 $r_v$와 수신한 $r_p$ 연접

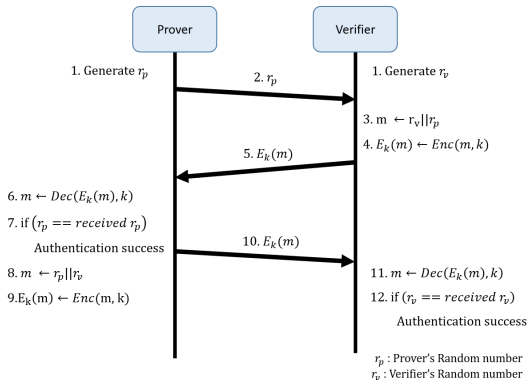


Fig. 8. CHAP Authentication process[15]

4. 검증자는 사전에 공유한 대칭키 $k$를 이용하여 $r_v \| r_p$ 암호화
5. 검증자는 $E_k(r_v \| r_p)$를 증명자에게 전송
6. 증명자는 수신한 $E_k(r_v \| r_p)$ 복호화
7. 증명자는 복호화한 값에서 얻어낸 $r_p$와 자신이 생성한 $r_p$를 비교하여 검증자 인증
8. 증명자는 $r_p$와 $r_v$ 연접
9. 증명자는 대칭키 $k$를 이용하여 $r_p \| r_v$ 암호화
10. 증명자는 검증자에게 $E_k(r_p \| r_v)$ 전송
11. 검증자는 수신한 $E_k(r_p \| r_v)$ 복호화
12. 검증자는 복호화한 값에서 얻어낸 $r_v$와 자신이 생성한 $r_v$를 비교하여 증명자 인증

### 3.4 키 교환

암호화를 수행하기 위해서는 암호화 과정에 사용되는 키를 사전에 공유해야 한다. 만약 키를 공유하는 과정에서 암호키가 노출되거나 예측되는 경우, 해당 암호 알고리즘은 안전성이 떨어지게 된다. 따라서 암호키를 공유할 때에는 안전한 키 교환 알고리즘을 이용하여야 한다.

대표적인 키 교환 알고리즘으로는 이산 대수 문제 방식을 이용하는 DH(Diffie-Hellman) 알고리즘이 있다. 하지만 DH 알고리즘의 경우 그룹별 비트 수가 1024-bit, 2048-bit 등 큰 연산량을 필요로 하기 때문에 자원이 제한적인 환경에서 사용하기에는 적합하지 않다. Table 6.는 보안강도에 따른 권장 키 길이를 나타낸다[16].

타원곡선을 기반으로 한 ECDH(Elliptic

Table 6. Key Size according to Security Strength[16]

| Security Strength (bit) | Discrete Logarithm Problem | | ECC (bit) |
|---|---|---|---|
| | public key (bit) | private key (bit) | |
| 112 | 2048 | 224 | 256 |
| 128 | 3072 | 224 | 256 |

-Curve Diffie-Hellman)는 DH에 비해 상대적으로 작은 양의 연산량을 필요로 하기 때문에, 드론과 같은 IoT 기기에서 키 교환을 하기에 적합하다. 또한 ECDH는 DH보다 저전력 IoT 기기에서 속도 및 전력소모량이 효율적이다[18]. Fig. 9.는 ECDH 키 교환 과정을 나타낸다.

1. Alice와 Bob이 각자의 개인키 $\alpha$, $\beta$ 생성
2. 기준이 되는 타원곡선 및 기준점 G 정의
3. Alice와 Bob은 타원곡선 상에서의 공개키 ($\alpha \cdot G$, $\beta \cdot G$) 공유
4. 공유한 공개키에 자신의 개인키를 연산하여 비밀키 생성

ECDH 알고리즘을 이용하여 키 교환을 한 경우, Alice와 Bob을 제외한 사용자가 비밀키를 알아내는 것은 정의된 타원곡선에서의 Base point G에 대한 전수 조사를 하는 경우에만 가능하다.

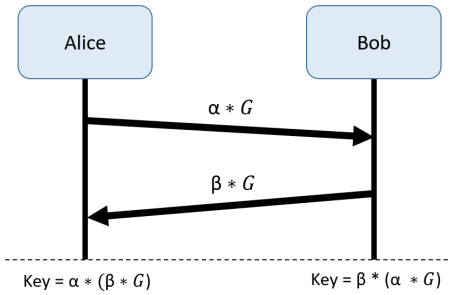


Fig. 9. ECDH key exchange algorithm[17]

## IV. 보안 설계 실험

### 4.1 실험 환경

드론의 사양과 비슷한 환경에서 모의실험을 진행하였다. 실험 전제 조건은 다음과 같다.

- 드론에는 양자 엔트로피 기반 난수 발생기를 추가한 모듈이 탑재되어 있다.
- 지상조종국과 드론 간의 첫 인증에 사용되는 암호키는 사전에 공유되어 있다.
- pixhawk 2를 대체하여 32bit Cortex-M4 CPU를 사용하였다.

Fig. 10.은 양자 난수를 이용한 인증 및 키 교환을 위한 실험 환경 구성도를 나타낸다. pixhawk에 양자 난수를 탑재한 초소형 칩을 부착하여 인증 및 키 교환을 수행한다.

Table 7.과 Table 8.은 양자 난수를 이용한 인증 및 키 교환을 위한 실험 장비 사양을 나타낸다.

Table 7.의 드론을 위한 실험 장비는 드론의 대표적인 모델인 pixhawk 2의 32bit Cortex-M4 CPU와 비슷한 성능을 가지는 초소형 칩으로, 모의실험 장비로 적합하다.

Table 9.은 실험에서 사용하는 양자 엔트로피 기

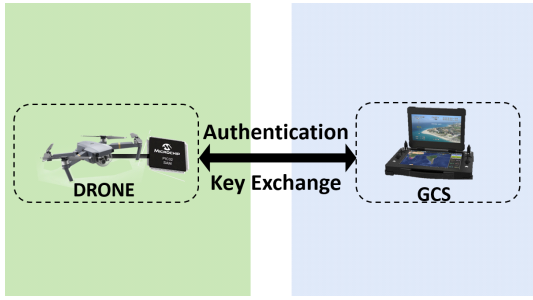


Fig. 10. Construction of experiment environment

Table 7. Specification of experiment equipment for Drone[20]

| Model name | SAMG55J19A |
|---|---|
| CPU | 32bit Cortex-M4 120 MHz |
| SRAM | 176 Kbytes |
| Flash | 512 Kbytes |
| Interface | USART |

Table 8. Specification of experiment equipment for GCS

| | GCS |
|---|---|
| OS | Ubuntu18.04 |
| CPU | i7 - 1065G7 |
| SRAM | 4096 bytes |
| Interface | USART |

Table 9. Specification of random number generation based quantum entropy[21]

| Model name | QEC |
|---|---|
| Operating temperature | -20℃~+85℃ |
| Voltage | 3.3V |
| Electric current | 800㎂ |
| Size | 3mm x 3mm |
| Passed test | NIST 90-B |
| Quality | 7.6bit efficiency over 8bit noise source samples |

반 난수 발생기의 사양을 나타낸다. 이 양자 엔트로피 기반 난수 발생기는 방사성 동위원소의 반감기를 이용하여 양자 엔트로피를 생성한다. 그 후 후처리 과정을 거쳐 HASH_DRBG를 이용하여 난수를 생성한다.

Table 10.는 실험에서 사용하는 notation을 나타낸다.

Table 10. Notation used in experiments

| Notation | Description |
|---|---|
| $QR_D$ | Quantum random number generated by drones |
| $R_{GCS}$ | Random number generated by GCS |
| $k$ | Pre-shared cryptographic keys used for first authentication |
| $KT_D$, $KT_{GCS}$ | Key token of drone and GCS |
| $QR_D \| R_{GCS}$ | A Value that are concatenated $QR_A$ and $QR_B$ |
| $E_k()$ | Encrypt with secret key $k$ |

## 4.2 인증

본 논문에서 사용하는 개체 인증 메커니즘은 난수와 대칭키 암호 알고리즘을 사용하는 Challenge-Response 양방향 인증 프로토콜 CHAP이다[15]. CHAP은 3단계 인증 방식을 사용하여 양방향 인증을 한다. 또한, 인증 개체들이 생성하는 난수와 대칭키를 사용하기 때문에, 두 개 이상의 서로 다른 인증 요인을 혼합하여 신원을 확인할 수 있는 강한 개체 인증 방식이다.

대칭키 알고리즘 중 LEA 알고리즘은 32-bit 환경에 최적화되어 설계되었고, Pixhawk 2의 CPU는 32-bit STM32F427 Cortex-M4F이므로

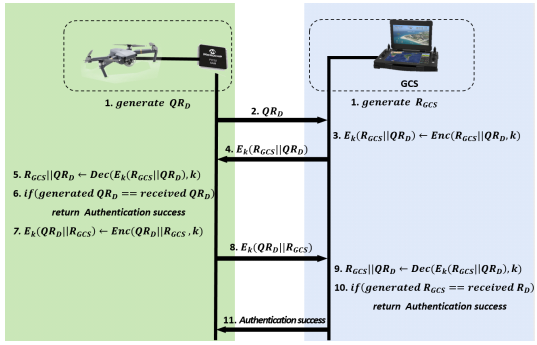


Fig. 11. Mutual Authentication Protocol

CHAP 인증 과정에서 LEA 알고리즘을 이용하면 다른 암호 알고리즘에 비해 훨씬 더 좋은 성능을 기대해 볼 수 있다[22]. Fig. 11.은 지상조종국과 드론 간의 양방향 개체 인증 프로토콜을 나타낸다.

1. 지상조종국은 난수 $R_{GCS}$, 드론은 탑재된 모듈의 양자 엔트로피 기반 난수 발생기를 이용하여 양자 난수 $QR_D$ 생성
2. 드론은 지상조종국에게 $QR_D$ 전송
3. $QR_D$를 수신한 지상조종국은 $R_{GCS}$와 $QR_D$ 연접 후, 사전 공유된 대칭키 $k$를 이용하여 암호화
4. 지상조종국은 드론에게 $E_k(R_{GCS} \| QR_D)$ 전송
5. 드론은 수신한 $E_k(R_{GCS} \| QR_D)$ 복호화
6. 드론은 복호화한 값에서 얻어낸 $QR_D$와 자신이 생성한 $QR_D$를 비교하여 지상조종국 인증
7. 드론은 $QR_D$와 복호화한 값에서 얻어낸 $R_{GCS}$를 연접 후, 대칭키 $k$를 이용하여 암호화
8. 드론은 지상조종국에게 $E_k(QR_D \| R_{GCS})$ 전송
9. 지상조종국은 수신한 $E_k(QR_D \| R_{GCS})$ 복호화
10. 지상조종국은 복호화한 값에서 얻어낸 $R_{GCS}$와 자신이 생성한 $R_{GCS}$를 비교하여 드론 인증
11. 지상조종국은 드론에게 인증 성공 메시지 전송

## 4.3 키 교환

지상조종국과 드론 간의 양방향 Challnge-Response 인증 과정을 마치면, 개체 간 송수신되는 데이터는 보안 서비스 중 개체 인증 서비스를 만족하게 된다.

양방향 인증이 완료된 후에 데이터 암호화를 위한

ECDH 키 교환 알고리즘을 진행한다. ECDH 키 교환 알고리즘은 키 토큰을 생성하는 과정에서 양자 난수를 사용한다. ECDH는 타원곡선 기반 알고리즘이므로 결과값은 좌표 형태인 (x,y)의 좌표 값이다. 이때 암호화에 사용되는 키는 x 좌표 값이다. Fig. 12.는 실험에서 사용된 ECDH 키 교환 알고리즘 구성도를 나타낸다.

1. 드론은 양자 난수 $QR_D$ 생성
2. 드론은 $QR_D$를 이용하여 키 토큰 $KT_D$ 생성 후 지상조종국에게 전송
3. 지상조종국은 $KT_D$ 수신 및 난수 $R_{GCS}$ 생성
4. 지상조종국은 $R_{GCS}$를 이용하여 키 토큰 $KT_{GCS}$ 생성 후 드론에게 전송
5. 드론은 $KT_{GCS}$와 $QR_D$를 이용하여 키 생성
6. 지상조종국은 $KT_D$와 $R_{GCS}$를 이용하여 키 생성

인증과 키 교환 과정을 모두 마친 드론은 서로 간 중요한 정보에 접근할 수 있게 된다. 이후 지상조종국과 드론 간 재인증이 필요한 경우, 마지막으로 키 교환 알고리즘을 통해 교환한 키를 인증에 사용한다.

Fig. 13.은 키 교환 과정 이후, 공유된 키를 이용하여 데이터를 주고받는 과정을 나타낸다. 해당 데이터는 기밀성이 보장된 안전한 데이터로, 허락되지 않은 사용자에게 데이터 유출이 되는 것을 방지할 수 있다.

### 4.4 결과 및 분석

수식 (7)은 인증 및 키 교환 과정의 성능 측정 계산식을 나타낸다.

$$T = \frac{1}{n}\sum_{i=1}^{n} t_i \ (t_i = t_{end} - t_{start}) \tag{7}$$

$t_i$ : $i$th operation time

$n$ : the number of experiments

Fig. 14.는 본 논문에서 제안하는 드론 시스템에서의 전체적인 보안 절차를 나타낸다.

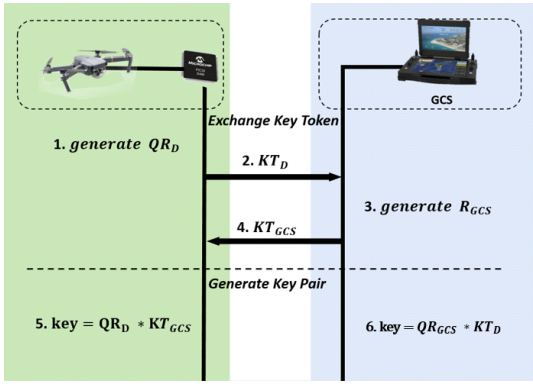


Fig. 12. Key Exchange Algorithm Diagram

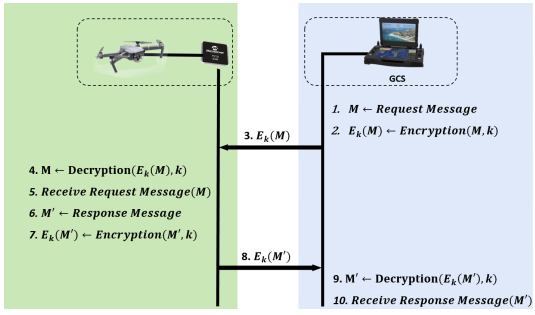


Fig. 13. Encryption for Data-in-transit

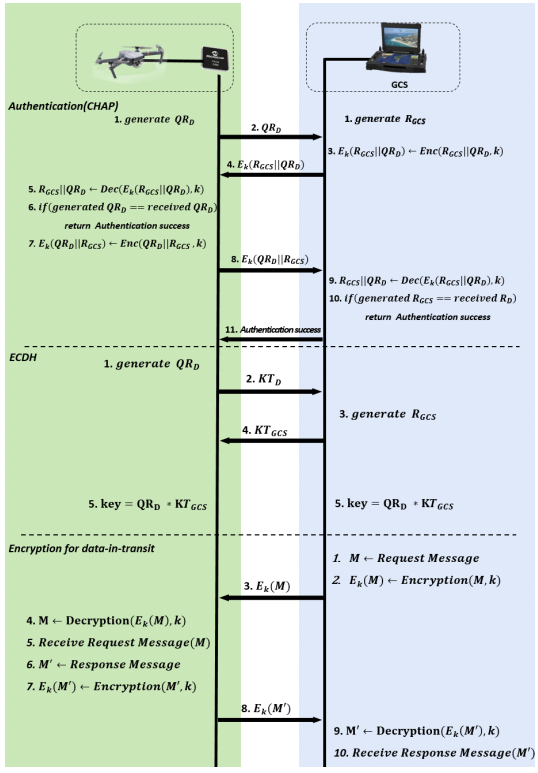


Fig. 14. Entire Procedure of Proposal

Table. 11. Performance measurement results

| Operation Function | Time |
|---|---|
| Generation Random Number | 66ms |
| Authentication(CHAP) | 88.6ms |
| Key Exchange(ECDH) | 881ms |
| LEA-128-CBC Decryption (255 bytes) | 2.5ms |
| LEA-128-CBC Encryption (255 bytes) | 2.5ms |

Table 11.은 난수 생성, 인증, 키 교환의 성능 측정 결과를 나타낸다. CPU 속도는 120MHz, Baud rate는 MAVLink 프로토콜에서 사용하는 것과 동일한 57600인 환경에서 100회 실험하여 평균을 낸 결과이다.

이 결과는 드론의 대표적인 모델과 비슷한 성능을 가진 장비로 실험하였으므로 유사 환경인 드론의 컨트롤러에 적용 가능할 것으로 보인다. 성능 측정 결과, 난수 생성은 66ms, 인증은 88.6ms, 타원 곡선 p-256 상에서의 키 교환은 881ms의 시간이 소요되었다. MAVLink v1, v2의 최대 페이로드 길이인 255bytes 데이터의 암복호화는 LEA-128 CBC 모드에서 2.5ms가 소요되었다. 이 결과에 따라 실제 드론에서의 인증과 키 교환 과정을 거친 후 암호화 과정에도 적용한다면 안전한 드론 시스템이 구성될 것을 기대해 볼 수 있다.

## V. 결 론

드론 시스템의 기술이 발전함에 따라 드론의 역할이 다양해지고 있지만, 드론 제어 데이터의 취약점 또한 증가하고 있다. 드론 기술의 순기능을 활용하기 위해서는 드론 제어 데이터의 보안 기술 개발 및 적용이 필수적이다.

본 논문에서는 드론 시스템에서 사용되는 메시징 프로토콜인 MAVLink v1, v2의 암호학적 취약점을 분석하여 기밀성, 무결성, 인증에 대한 취약점을 확인하였다. 이는 MAVLink 프로토콜을 사용하는 드론에서의 보안 취약점과 직결된다. 이러한 취약점으로부터 데이터를 보호하기 위해 유사 환경에서, 드론과 지상국 간의 인증 및 키 교환, 데이터 암복호화를 위한 실험을 진행하였고, 각 보안 사항에 대한 소요 시간을 측정하였다.

앞으로도 드론 기술은 점진적으로 발전할 것으로 전망된다. 따라서 드론 데이터의 보안 취약점을 위한 데이터 보안 기술에 대한 연구는 지속적으로 이루어져야 한다.

## References


[1] “A Study on the Domestic and Foreign Trends of Drone Industry Based on 4th Industrial Revolution,”, https://www.gb.go.kr/Main/open_contents/section/economy/page.do?mnu_uid=2874&BD_CODE=bbs_gongji&cmd=2&B_NUM=70035301&B_STEP=70035300&V_NUM, Feb. 2021.

[2] Joseph A. Marty, “Vulnerability Analysis of the MAVLink Protocol for Command and Control of Unmanned Aircraft,” Mar. 2014.

[3] You-sung Kang, Ju-han Kim, Kun-woo Kim and Seung-Gwang Lee. “Security Requirements for Drone-based Services,” TTAK.KO-12.0317, June. 2016.

[4] Yong-hyun Yoon, Kyung-ryun Oh, Sung-hoon Shin, Sang-seop Lee and Dong-geun Lee, “Industry Guide for Global Technical Regulations on Drone(Unmanned Aircraft),” KATS(Korean Agency for Technology and Standards), KSA(Korea Standard Association), 2018-2, Jan. 2018.

[5] Anis Koubaa, Azza Allouch, Maram Alajlan, Yasir Javed, Abdelfettah Belghith, and Mohamed Khalgui, “Micro Air Vehicle Link (MAVLink) in a Nutshell: A survery,” IEEE Access, vol. 7, pp. 87658-87680, June. 2019.

[6] GitHub, “mavlink” www.github.com/mavlink, Feb. 2021.

[7] Quyuh Dang, “Recommendation for Applications Using Approved Hash Algorithms,”, 800-107 Rev 1, Aug. 2012.

[8] Seong-Min Cho, Seung-Hyun Seo, “Status of cryptographic technologies

applied to drone security," Journal of the Korea Institute of Information Security & Cryptology, 30(2), pp. 11-19, Apr. 2020.
[9] Myoung-su Kim, Il-sun You and Kang-bin Yim, "Analysis of Fragility and Response Technology of Unmanned Mobile Drone," Journal of the Korea Institute of Information Security & Cryptology, 30(2), pp. 49-57, Apr. 2020.
[10] N. Buthcer and A. Stewart "Securing the MAVLink Communication for Unmanned Aircraft Systems," CSSE14-02, University of Appalachian State and University of Auburn, 2014
[11] TTA, "TTAK.K0-12.0235/R2", Dec. 2020.
[12] SEONG-MIN CHO, EUNGI HONG, and SEUNG-HYUN SEO, "Random Number Generator Using Sensors for Drone," IEEE Access, vol. 8, pp. 30343-30354, Feb. 2020.
[13] Jungmin Park, Seongjoon Cho, Taejin Lim, and Mark Tehranipoor, "QEC: A Quantum Entropy Chip and Its Applications," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, pp. 1471-1484, June. 2020.
[14] Rakel Haakegaard, Joanna Lang, "The Elliptic Curve Diffie-Hellman (ECDH)," Dec. 2015.
[15] Marin Feldhofer, Sandra Dominikus, and Johannes Wolkersstorfer, "Strong Authentication for RFID Systems Using the AES Algorithm," CHES 2004, vol. 3156, Aug. 2004.
[16] KISA, "Password Algorithm and Key Length Usage Guide," December, 2018
[17] SECG, "SEC 1: Elliptic Curve Cryptography", Apr. 2009.
[18] Tarun Kumar Goyal, Vineet Sahula, "Light Weight Security Algorithm for Low Power IoT Devices," 2016 ICACCI (International Conference on Advances in Computing, Communications and Informatics), Sep. 2016.
[19] PX4, "Pixhawk", https://px4.io/, Feb. 2021.
[20] Microchip, "SMART SAM G55G SAM G55J Data sheet", May. 2016.
[21] EYL, "ENTROPY CHIP: GAME CHANGER FOR IoT SECURITY," Dec. 2016
[22] Si-hoon Moon, Min-woo Kim and Tae-kyung Kwon, "Trends in lightweight cryptographic technology for IoT communication environment," The Journal of The Korean Intitute of Communication Sciences, 33(3), pp. 80-86, Feb. 2016.

## 〈저자소개〉

김 태 완 (Tae-Wan Kim) 학생회원
2021년 2월: 국민대학교 정보보안암호수학과 졸업
2021년 2월~현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 정보보호, 네트워크보안, IoT보안, 5G/6G 보안

이 세 윤 (Se-Yoon Lee) 학생회원
2021년 2월: 국민대학교 정보보안암호수학과 졸업
2021년 2월~ 현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 정보보호, 네트워크보안, IoT보안, 5G/6G 보안

정 서 우 (Seo-Woo Jung) 학생회원
2021년 2월: 국민대학교 정보보안암호수학과 졸업
2021년 2월~현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 정보보호, 네트워크보안, IoT보안, 5G/6G 보안

위 한 샘 (Han-saem Wi) 학생회원
2016년 2월: 국민대학교 수학과 졸업
2018년 8월: 국민대학교 금융정보보안학과 석사 졸업
2018년 9월~현재: 국민대학교 금융정보보안학과 박사과정
〈관심분야〉 정보보호, 네트워크보안, IoT보안, 5G/6G 보안

이 옥 연 (Ok-yeon Yi) 종신회원
1988년 2월: 고려대학교 수학과 졸업
1990년 2월: 고려대학교 수학과 석사
1996년 8월: Univ. of Kentucky 박사
〈관심분야〉 5G/6G 보안, 위성통신 보안, KCMVP