creative commons
COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

석사학위논문

드론 조종기 및 클라우드 데이터 분석을 통한 드론 포렌식 연구

DJI 및 Parrot 드론 대상의 분석

HANSUNG UNIVERSITY

2026년

한 성 대 학 교 대 학 원

융 합 보 안 학 과

융 합 보 안 전 공

김 한 결

HANSUNG
UNIVERSITY

석사학위논문
지도교수 박명서

드론 조종기 및 클라우드 데이터 분석을 통한 드론 포렌식 연구

DJI 및 Parrot 드론 대상의 분석

A Study on Drone Forensics through Drone Controller and Cloud Data Analysis

HANSUNG UNIVERSITY

2026년 6월 일

한 성 대 학 교 대 학 원

융 합 보 안 학 과

융 합 보 안 전 공

김 한 결

석사학위논문
지도교수 박명서

드론 조종기 및 클라우드 데이터 분석을 통한 드론 포렌식 연구

DJI 및 Parrot 드론 대상의 분석

A Study on Drone Forensics through Drone Controller and Cloud Data Analysis

HANSUNG UNIVERSITY

위 논문을 공학 석사학위 논문으로 제출함

2026년 6월 일

한 성 대 학 교 대 학 원

융 합 보 안 학 과

융 합 보 안 전 공

김 한 결

김한결의 공학 석사학위 논문을 인준함

2026년 6월 일

심사위원장 서화정 (인)

심 사 위 원 박명서 (인)

심 사 위 원 석병진 (인)

HANSUNG UNIVERSITY

국 문 초 록

드론 조종기 및 클라우드 데이터 분석을 통한 드론 포렌식 연구

한 성 대 학 교 대 학 원
융 합 보 안 학 과
융 합 보 안 전 공
김 한 결

드론은 민간 및 산업 전반에서 활용 범위가 확대됨에 따라 다양한 편의성을 제공하는 동시에, 새로운 범죄 수단으로도 악용되고 있다. 이러한 드론 범죄는 조종자가 현장과 분리된 환경에서 행위를 수행할 수 있다는 특성으로 인해 비행 로그 및 미디어 파일 등 디지털 증거에 기반한 포렌식 분석이 필수적이다. 기존의 드론 포렌식 연구는 주로 드론 기체에 저장된 데이터 분석에 집중되어 왔으나, 실제 수사 환경에서는 조종자 식별과 행위 추적을 위한 조종자 중심의 디지털 증거 분석이 중요하다. 특히 클라우드 기반으로 조종자 데이터가 관리되는 최신 드론 환경에서는 제조사 서버에 저장된 원격 데이터에 대한 분석이 요구된다. 이러한 배경 속에서 본 연구는 드론 시장에서 높은 점유율을 보유한 DJI와 Parrot 드론을 대상으로, 모바일 조종 어플리케이션과 임베디드 조종기를 포함하는 조종자 중심의 디지털 포렌식 분석 방법을 제안한다. 조종기의 로컬 저장소와 클라우드 서버로부터 조종자 관련 데이터를 수집하고 분석함으로써, 드론 범죄 재구성을 위한 통합 분석 절차를 제시한다.

【주요어】 Digital Forensics, Drone Forensics, DJI Drone, Parrot Drone

목 차

제 1 장 서 론 ····· 1

제 1 절 배경 ····· 1
제 2 절 관련 연구 ····· 2
제 3 절 연구 기여 ····· 5

제 2 장 드론 조종기 및 클라우드 데이터 수집 방법론 ····· 7

제 1 절 데이터 수집 ····· 7
1) 조종기 로컬 저장소 접근 ····· 7
2) 네트워크 패킷 재구성 ····· 8
제 2 절 분석 방지 우회 ····· 9
1) 안티 디버깅 ····· 9
2) 패킹 ····· 10
3) SSL 인증서 고정 기법 ····· 11

제 3 장 데이터 수집 및 분석 기술 ····· 12

제 1 절 데이터 수집 ····· 12
1) 드론 조종기 ····· 12
2) 드론 클라우드 ····· 13
제 2 절 드론 조종기 데이터 분석 ····· 21
1) 드론 기체 정보 ····· 28
2) 조종자 정보 ····· 32
3) 비행로그 파일 ····· 35
4) 미디어 파일 ····· 39
5) 어플리케이션 로그 파일 ····· 43
제 3 절 클라우드 데이터 분석 ····· 45
제 4 절 비행로그 파일 복호화 ····· 53

HANSUNG UNIVERSITY

제 4 장 드론 포렌식 통합 분석 ····· 57

제 1 절 드론-조종자 상관성 분석 ····· 57
1) 드론 시리얼 번호 식별 ····· 57
2) 상관성 분석 ····· 59

제 2 절 6W 관점의 드론 범죄 재구성 ····· 60

제 5 장 결 론 ····· 64

참 고 문 헌 ····· 65

ABSTRACT ····· 69

HANSUNG UNIVERSITY

표 목 차

[표 3-1] ADB를 통한 조종기 데이터 수집 경로 ····· 13
[표 3-2] DJI 클라우드의 주요 API 식별 결과 ····· 14
[표 3-3] Parrot 클라우드의 주요 API 식별 결과 ····· 19
[표 3-4] 드론 조종 어플리케이션 데이터 분석 결과 ····· 21
[표 3-5] 드론 기체 관련 데이터 분류 ····· 28
[표 3-6] 드론 조종자 관련 데이터 분류 ····· 32
[표 3-7] 드론 제조사별 비행로그 파일에 저장된 주요 정보 ····· 35
[표 3-8] Parrot Freeflight 6의 비행로그 상세 파일 분석 결과 ····· 38
[표 3-9] 미디어 파일 분석 결과 분류 ····· 40
[표 3-10] Parrot Freeflight 6의 미디어 파일 메타데이터 분석 결과 ····· 42
[표 3-11] 드론 조종 어플리케이션 로그 파일 분석 결과 ····· 43
[표 3-12] DJI 클라우드 API의 응답 데이터 분석 결과 ····· 46
[표 3-13] Parrot 클라우드 API의 응답 데이터 분석 결과 ····· 51
[표 4-1] 6W 관점에서 분석 데이터의 분류 결과 ····· 61

HANSUNG UNIVERSITY

그 림 목 차

[그림 2-1] DJI 어플리케이션의 디버거 부착 방지 기술 우회 절차 ····· 10
[그림 3-1] DJI Cloud 서버를 통한 로그인 API 호출 절차 ····· 16
[그림 3-2] Sign-Mc 생성 절차 ····· 17
[그림 3-3] Parrot Cloud의 로그인 API token 생성 알고리즘 도식화 ····· 20
[그림 3-4] DJI GO의 dji.db 분석을 통한 드론 정보 식별 결과 ····· 29
[그림 3-5] DJI GO4의 mmkv.default 분석을 통한 드론 정보 식별 결과 ··· 30
[그림 3-6] Parrot Freeflight 6의 activation.xml 분석을 통한 드론 정보 식별 결과 31
[그림 3-7] Parrot Freeflight 6의 ff6Prefs.xml 분석을 통한 드론 정보 식별 결과 ··· 31
[그림 3-8] DJI GO의 비행로그 동기화 로그 파일 내 조종자 정보 식별 결과 ··· 33
[그림 3-9] flysafe_license_unlock.db 파일 분석을 통한 조종자 정보 식별 결과 ···· 34
[그림 3-10] Parrot Freeflight 6의 activation.xml 분석을 통한 조종기 시리얼 번호 식별 결과 34
[그림 3-11] Parrot Freeflight 6의 비행로그 요약 파일 분석 결과 ····· 37
[그림 3-12] Parrot Freeflight 6의 비행로그 상세 파일 분석 결과 ····· 37
[그림 3-13] Parrot Freeflight 6의 비행로그 시각화 결과 ····· 39
[그림 3-14] DJI GO4에서의 동영상 파일과 info 파일 분석 결과 ····· 42
[그림 3-15] DJI GO4의 ERROR_POP_LOG 분석 결과 ····· 44
[그림 3-16] DJI Fly의 AeroScopeLogic 로그 파일 분석 결과 ····· 44
[그림 3-17] DJI 클라우드의 API-3 응답 데이터 분석 결과 ····· 47
[그림 3-18] DJI 클라우드의 API-5 응답 데이터 분석 결과 ····· 48
[그림 3-19] DJI 클라우드의 API-6 응답 데이터 분석 결과 ····· 48
[그림 3-20] DJI 드론 조종 어플리케이션을 통한 드론 시리얼 번호 식별 결과 ····· 49
[그림 3-21] 조종자의 계정에 바인딩된 기기 정보 조회 ····· 50
[그림 3-22] 드론에 바인딩된 계정 및 조종기 정보 조회 ····· 51
[그림 3-23] Parrot 클라우드 API를 통해 수집한 비행로그 파일 분석 결과 ··· 52
[그림 3-24] DJI 드론 앱의 TXT 비행로그 파일 구조 ····· 54
[그림 3-25] OpenAPI를 활용한 DJI Fly의 비행로그 복호화 절차 ····· 55
[그림 3-26] DJI 드론 앱의 TXT 비행로그 파일에 대한 복호화 시각화 결과 비교 ··· 56
[그림 4-1] DJI 드론 기체에서의 시리얼 번호 식별 결과 ····· 57
[그림 4-2] Parrot 드론 기체에서의 시리얼 번호 식별 결과 ····· 58
[그림 4-3] DJI 드론 환경에서의 상관성 분석 과정 ····· 59
[그림 4-4] Parrot 드론 환경에서의 상관성 분석 ····· 60
[그림 4-5] 6W 관점에서의 드론 포렌식 통합 분석 결과 ····· 63

수 식 목 차

[수식 3-1] API-2의 password 필드 암호화 알고리즘 ····· 16
[수식 3-2] API-9의 signature 생성 알고리즘 ····· 18
[수식 3-3] API-10, 11의 signature 생성 알고리즘 ····· 18
[수식 3-4] DJI 앱 로그 파일 복호화 알고리즘 ····· 45

HANSUNG UNIVERSITY

제 1 장 서론

제 1 절 배경

드론은 무인항공기(Unmanned Aerial Vehicle)를 의미하며, 통신·비행 기술의 고도화와 함께 민간 및 다양한 산업 분야에서 활용이 급속히 확산되고 있다. 드론은 원격 조종 또는 자동화된 비행 제어를 통해 조종자가 드론 기체와 물리적으로 분리된 위치에서 임무를 수행할 수 있게 함으로써, 인력 접근이 제한되거나 위험도가 높은 환경에서도 효율적인 정보 수집 및 작업 수행을 가능하게 한다. 특히 고해상도 카메라 및 다양한 센서를 탑재할 수 있어 항공 영상 촬영, 지리 탐색, 시설물 점검, 재난·재해 대응 등에서 비용 대비 높은 관측 효율성과 신속한 현장 대응성을 제공한다. 더 나아가, 드론은 시간·공간적 제약을 완화하고 작업 단위를 소형화하여, 기존 항공 촬영이나 지상 인력 중심의 운영 대비 운용 비용을 절감한다. 그러나 이러한 효용성의 확대와 함께 드론을 악용한 다양한 위협 시나리오와 범죄 양상 또한 다변화되고 있다. 드론이 공항 또는 비행 금지 구역 인근에서 무단으로 비행한 경우 항공기 운항 안전을 저해하고, 충돌 위험 또는 운항 중단과 같은 심각한 사회·경제적 피해를 야기할 수 있다. 또한 드론은 소형·기동성·원격성이라는 특성으로 인해 물품 운반에 악용되기 용이하며, 이를 통한 밀수·밀매 또는 제한 물품 투하와 같은 범죄가 보고되고 있다. 아울러 공중 촬영 기능은 군사 시설에 대한 정찰 또는 제 3자에 대한 불법 촬영 등을 통해 국가와 사회 안전 측면에서 법적·윤리적 문제를 초래할 수 있다. 이와 같은 드론 범죄는 전통적 범죄와 달리 행위 주체가 현장으로부터 분리될 수 있고, 드론의 비행 로그, 미디어, 통신 기록 등 디지털 흔적에 대한 의존도가 높아 사건의 행위 재구성과 책임소재 규명에서 디지털 포렌식의 역할이 중요하다. 이러한 배경 속에서 드론을 대상으로 한 다양한 디지털 포렌식 연구가 수행되어 왔으나, 기존 연구는 주로 드론 기체의 저장매체 및 비행 로그 분석을 중심으로 진행되었다. 드론 범죄에서 드론의 비행 경로와 시각, 고

도, 카메라 촬영 등 기체의 행위는 사건을 입증하는 핵심 근거가 될 수 있다. 그러나 실제 수사에서 범죄 사건을 재구성하기 위해서는 드론 기체 데이터 뿐 아니라 조종자를 식별하고, 조종자의 행위를 추적하는 조종자 중심의 정황 증거가 필수적이다. 특히 DJI와 Parrot의 드론과 같이 드론 조종 어플리케이션을 통해 클라우드 서버에 조종자의 데이터가 관리되는 환경에서는 기존에 논의되지 않은 드론 제조사의 원격지 서버에 대한 접근 기술이 요구된다.

본 연구에서는 드론 시장에서 높은 점유율을 보유한 DJI와 Parrot의 드론을 대상으로 드론 포렌식 연구를 수행한다. 분석 대상으로는 DJI의 조종 어플리케이션인 DJI GO, GO4, Fly, Pilot과 Parrot의 FreeFlight 6를 포함하며, 대상 기체로는 DJI Phantom 3/4, Mini 2, Mavic 2 Enterprise와 Parrot ANAFI를 선정하였다. 또한 조종 환경의 특성을 고려하여, 모바일 환경에서 동작하는 드론 조종 어플리케이션뿐만 아니라 해당 어플리케이션이 구동되는 임베디드 조종기인 DJI Smart Controller까지 분석 대상으로 포함하는 조종자 중심의 포렌식 분석을 수행한다. 이를 위해 조종기를 대상으로 로컬 저장소에 존재하는 데이터와 클라우드 서버에 저장·동기화된 조종자의 클라우드 데이터를 수집한다. 이후 수집된 데이터를 드론 기체 정보, 조종자 정보, 비행로그 파일, 미디어 파일, 어플리케이션 로그 파일로 분류하여 분석을 수행하고, 이를 기반으로 드론 범죄에 대한 통합 분석을 수행한다.

본 논문의 구성은 다음과 같다. 2장에서는 연구 수행 과정에서 적용한 데이터 수집 및 보안 기술 우회에 대한 기술적 방법론을 설명한다. 3장에서는 드론 조종기의 로컬 저장소와 클라우드 서버로부터 데이터를 수집하는 방법과, 수집된 데이터를 분석한 결과를 제시한다. 4장에서는 분석된 결과를 기반으로 통합 분석을 수행하여 드론 범죄를 재구성한다. 5장에서는 연구의 결론으로 마무리한다.

제 2 절 관련 연구

드론 포렌식 분야에서는 다양한 제조사의 드론을 대상으로 디지털 증거를 수집하기 위한 기술 연구가 진행되었다.

1) 드론 기체 및 비행로그 중심의 연구

Devon R. Clark 등 4명은 DJI Phantom III의 DAT 비행로그를 분석하기 위한 오픈소스 도구인 DROP(DRone Open source Parser)을 제시하였다. 해당 연구는 암호화된 DAT 비행로그를 복호화하여 GPS 위치, 배터리, 비행시간 데이터를 획득하였으며, 모바일 기기와 드론 간 데이터 상관성 분석 방법을 제안하였다. Ankit Renduchintala 등 4명은 DJI Phantom 4 및 Yuneec Typhoon H 드론을 대상으로 다양한 비행 시나리오를 구축하고, 비행로그 파일을 분석하여 시각화하였다. Yousef 등 3명은 4개의 상업용 드론을 대상으로 시나리오 기반 비행을 통해 데이터를 생성하고, FTK Imager, Autopsy, CsvView 등의 포렌식 도구를 활용하여 드론 메모리, 미디어 파일, 비행로그 데이터를 분석하였다. Kumar 등 2명은 DJI Phantom 4 Pro, Parrot Bebop 2, Yuneec Typhoon H 드론의 비행로그를 DatCon, FlyLog Converter Tool, Google Earth Pro 등의 공개 도구를 통해 분석하고 비행 경로를 재구성하였다. Sukyung Cho 등 5명은 DJI Mavic Pro, Mavic Air2, Mini2를 대상으로 DAT 비행로그 파일의 구조 유형을 제안하고 저장 구조 및 암호화 방식을 분석하였으며, 이를 분석하기 위한 포렌식 도구를 개발하였다. YeoHoon Yoon 등 2명은 DJI 드론의 삭제된 비행 기록 파일의 특성을 식별하고, 파일 카빙을 통한 복구 가능성을 15가지 드론 모델에 대해 분석하였다. Dong-Min Woo 등 2명은 오픈소스 드론 운영체제 PX4의 로그 파일 생성 과정 및 암호화 방식을 분석하고, 암호화 알고리즘과 시점 변경을 통한 안정성을 확인하였다. Seyoung Baik 등 2명은 오픈소스 드론을 대상으로 한 라이브 포렌식 도구인 LiPFo(Live-PX4-Forensic)를 개발하고, 이를 활용한 포렌식 방법론을 제시하였다. Hana Bouafif 등 3명은 Parrot AR 드론에 대한 포렌식 분석을 진행하였다. Android 기기에서 AR.Freeflight 어플리케이션을 통해 미디어 및 비행 데이터를 생성하고, Telnet·FTP·USB 연결을 통해 드론에 접근하여 미디어 데이터의 Exif 메타데이터와 내·외장 저장장치의 아티팩트를 분석하였다. Hyunji Moon 등 5명은 DJI Phantom 4 Pro를 사용하여 모터

전류 값과 조종기 방향 값을 분석함으로써 외부 힘에 의한 비정상 비행을 탐지하는 디지털 포렌식 방법론을 개발하였다.

2) 드론 조종 어플리케이션 및 모바일 환경 중심의 분석 연구

Chi-Cheng Yang 등 3명은 DJI Spark와 Mavic Air를 대상으로 비행로그 파일 분석과 모바일 아티팩트 수집을 수행하였으며, DJI Assistant 2와 Airdata UAV를 활용하여 드론의 DAT 비행로그를 분석하였다. Stankovic 등 3명은 DJI Mini 2의 SD카드 및 Android·iOS 기기에 저장된 데이터에 대한 포렌식 분석을 수행하였다. 다양한 범죄 시나리오를 설계하고 Cellebrite, AXIOM 등의 포렌식 도구를 활용하여 DJI Fly 어플리케이션 아티팩트로부터 개인식별정보(PII)를 복구하였으며, 미디어 파일의 Exif 메타데이터 분석을 통해 드론 카메라 시리얼 번호를 추출하였다. Khalifa Al-Room 등 7명은 DJI Phantom 3·4, Parrot Bebop, Syma X5C 등 다양한 상업용 드론에 대한 포렌식 분석을 진행하였다. 드론 기체의 내·외장 저장장치로부터 데이터를 획득하고, DJI 드론에 대해서는 DJI GO 어플리케이션 분석을 함께 수행하였으며, DAT 및 TXT 비행로그 경로 분석과 미디어 데이터의 Exif 메타데이터 분석을 진행하였다. Ziyu Zhao 등 3명은 DJI Mavic 2 Pro와 DJI GO4 어플리케이션을 중심으로 포렌식 연구를 수행하였으며, 드론의 내·외부 메모리로부터 데이터를 수집하는 파서(DRDP)를 제안하고 비행로그 파일 분석을 수행하였다. 본 연구에서는 해당 연구의 TXT 비행로그 분석 결과를 기반으로 다양한 드론 조종 어플리케이션에 대한 적용 가능성을 식별하였다. Husnjak 등 4명은 DJI Mavic Air와 DJI GO4 어플리케이션을 대상으로 5단계 수사 방법론을 적용하여 비행 기록, 미디어 파일 및 관련 아티팩트를 식별하고 분석하였다. Youngwoo Lee 등 4명은 비행로그 복호화 여부에 따라 DJI 드론을 분류하는 연구를 진행하였으며, 내·외부 저장장치에서 미디어 데이터와 비행로그 파일을 획득하여 Airdata, Phantomhelp, CsvView 도구를 통해 분석하였다. Fahad E. Salamh 등 3명은 DJI Matrice 210과 Phantom 4를 대상으로 포렌식 분석을 진행하였다. DatCon, CsvView, Autopsy, Cellebrite 도

구를 사용하여 드론의 내부 및 외부 SD카드에서 개인식별정보(PII)를 복구하였다.

3) 드론 임베디드 조종기 중심의 연구

Sungwon Lee 등 3명은 디지털 포렌식 관점에서 DJI의 임베디드 조종기인 DJI Remote Controller를 분석하였다. 해당 연구에서는 상용 도구를 통해 조종기에 내장된 DJI Fly 어플리케이션 데이터를 수집하고, 이를 7W (Who, Which, When, Where, Where to Where, What)관점에서 분석을 수행하였다. 본 연구에서는 선행 연구의 7W 형태의 분석 방식에 기반하여, Where to Where를 하나의 드론의 비행 경로로 정의하여 6W 관점에서 분석하였다. Sehoon Lee 등 4명은 DJI 드론을 대상으로 DUML(DJI Universal Markup Language) 프로토콜을 통해 드론 기체를 파손하지 않고 내부 메모리에 접근하는 기술적 방법을 제시하였다.

앞선 선행 연구들은 드론 기체의 저장매체, 비행로그 파일, 모바일 어플리케이션 아티팩트 등 디바이스 로컬 데이터 중심의 분석에 집중되어 있다. 로컬 데이터는 조종기 초기화, 데이터 삭제, 잠금 등의 상황에서 수집이 제한될 수 있다는 본질적인 한계를 갖는다. 이에 비해 본 연구는 조종자의 계정을 통해 제조사 클라우드 서버에 저장·동기화된 데이터를 추가적인 증거 계층으로 활용한다. 클라우드에 저장된 조종자–드론 바인딩 정보, 비행 메타데이터, 기기 식별자 등은 로컬 데이터가 삭제된 이후에도 서버에 잔존하는 특성을 지니며, 특정 조종기에 종속되지 않고 접근 가능하다는 이점을 갖는다. 본 연구에서는 이러한 클라우드 기반 증거를 로컬 아티팩트와 연계한 통합 분석을 통해, 기존 연구가 다루지 못한 조종자 중심의 드론 범죄 재구성 절차를 제시한다.

제 3 절 연구 기여

본 연구는 드론 조종자를 중심으로 한 데이터 수집과 분석을 수행하고, 로

컬 및 클라우드 데이터를 연계한 통합 분석을 통해 드론 범죄 사건을 재구성한다. 본 연구의 주요 기여는 다음과 같다.

첫째, 드론 조종 어플리케이션이 동작하는 모바일 환경과 임베디드 조종기 환경을 대상으로 포괄적인 로컬 데이터 분석을 수행하였다. 이를 통해 각 환경 및 어플리케이션별 데이터 저장 경로와 주요 디지털 증거 항목을 식별하였으며, 저장된 암호화 데이터를 복호화하여 디지털 포렌식 분석에 활용 가능한 형태로 확보하였다.

둘째, 기존 연구에서 충분히 다루어지지 않았던 클라우드 기반 드론 포렌식 분석을 수행하였다. 조종자 계정을 통해 생성되어 제조사별 클라우드 서버에 저장되는 데이터는 조종기의 초기화 또는 로컬 데이터 삭제 이후에도 서버에 잔존하는 특성을 가지며, 디지털 포렌식 관점에서 중요한 증거 가치를 가진다. 본 연구에서는 드론 조종 어플리케이션과 클라우드 서버 간의 네트워크 통신을 분석하고, 어플리케이션이 서버로 전송하는 요청 패킷을 재구성함으로써 클라우드 데이터를 수집할 수 있는 기술적 방법을 제시한다. 또한 클라우드 통신 과정에서 적용되는 암호화, 재전송 방지 기법, 캡차 등 보안 메커니즘에 대한 역공학 분석을 수행하고, 이를 우회하여 데이터를 수집할 수 있는 방법을 제안한다.

셋째, 조종기의 로컬 저장소와 클라우드 서버로부터 획득한 데이터를 연계한 통합 분석 방법을 제시한다. 이를 위해 드론 기체와 조종기 간의 연관성을 1차적으로 검증하고, 이후 수집된 다양한 디지털 증거를 종합적으로 분석함으로써 드론 범죄 사건을 재구성하는 절차를 제시한다.

제 2 장 드론 조종기 및 클라우드 데이터 수집 방법론

제 1 절 데이터 수집

1) 조종기 로컬 저장소 접근

드론을 조종하는 조종기는 어플리케이션이 내장되는 단독 조종기 형태와 모바일 기기를 조이스틱이 존재하는 드론 조종 장치에 연결하여 조종기로써 사용하는 형태로 분류된다. 본 연구에서 분석 대상으로 선정한 두 유형의 조종기는 모두 Android 운영체제에서 동작한다. 조종자는 어플리케이션을 구동하여 드론을 조종할 수 있으며, 이러한 과정에서 데이터가 조종기의 로컬 저장소에 저장된다. 데이터가 저장되는 경로는 어플리케이션이 동작하는 기기이며, 결과적으로 Android 환경의 데이터 파티션과 유저 파티션에 데이터가 분산되어 저장된다. 디지털 포렌식 관점에서 로컬 저장소에 저장된 데이터는 드론 조종자와 드론의 행위뿐만 아니라, 드론 범죄 사건을 재구성할 수 있는 단서를 제공한다. 따라서 본 연구에서는 데이터 수집을 위해 드론 조종기의 로컬 저장소에 접근하였다.

조종기의 형태와 상관 없이 Android 환경은 ADB(Android Debug Bridge)를 통해 안드로이드 Shell에 접근할 수 있다. 이를 위해 선행적으로 Android의 개발자 옵션 탭에서 USB 디버깅을 활성화하여 ADB 포트를 개방해야 한다. 본 연구에서는 분석 PC와 조종기를 USB 포트를 통해 연결한 후 ADB의 ‘adb shell’ 및 ‘adb pull’ 명령어를 사용하여 로컬 저장소 탐색 및 데이터 수집을 진행하였다. 그러나 앞선 ADB를 통한 접근은 일반적인 Android 환경에서 로컬 저장소 탐색에 한계점이 존재한다. Android 환경에서는 사용자의 권한 수준에 따른 접근제어가 적용되며, 이로 인해 데이터 파티션에 접근이 불가능하다. 이러한 문제를 해결하기 위해 본 연구에서는 루팅된 모바일 기기를 활용하였다. 루팅된 기기에서는 ‘adb shell’을 통한 접속 시 ‘su’ 명령어를 통해 권한 상승이 가능하며, 결

과적으로 데이터 파티션에 접근할 수 있다. 단독 조종기 형태는 공개된 취약점을 통해 데이터 수집을 진행하였다. 본 연구 대상인 단독 조종기는 DJI SC (DJI Smart Controller)로 공개된 취약점이 존재한다. 해당 조종기는 업데이트를 지원하지 않아 연구 시점에서도 취약점이 유효하게 동작하였다. 이러한 DJI SC의 취약점은 조종기에서 동작하는 djilink 바이너리에 존재하는 OS Command Injection 유형으로, 루트 권한으로 동작하는 djilink 프로세스에 명령어를 주입하여 관리자 권한을 통한 임의의 명령어 실행이 가능하다. 본 연구에서는 이러한 취약점을 활용하여 데이터 파티션에 존재하는 앱 데이터를 유저 파티션으로 복사 후, ADB를 통해 데이터를 수집하였다.

2) 네트워크 패킷 재구성

드론 조종 어플리케이션은 제조사의 클라우드 서버와의 통신을 통해 원격지 서버에 조종자의 정보를 저장하고, 비행로그 파일과 같은 데이터를 동기화한다. 이러한 클라우드 데이터를 수집하기 위해선 어플리케이션이 클라우드 서버로 전송하는 네트워크 패킷을 재구성해야 한다. 본 연구에서는 PC 환경에 모바일 기기를 USB 포트를 통해 연결한 후, 프록시 도구인 Fiddler Everywhere를 통해 어플리케이션과 클라우드 서버의 네트워크 통신을 분석하였다. 이러한 네트워크 통신은 제조사별로 동일한 통신 내용을 가지기 때문에, DJI SC에 대한 별도의 통신 분석은 수행하지 않았다.

프록시 도구를 통해 확인 가능한 주요 정보는 API Endpoint, HTTP Method·Header·Body·Parameter 정보이다. DJI와 Parrot의 드론 조종 어플리케이션은 HTTP/HTTPS 프로토콜을 통해 네트워크 통신을 수행하며, 응답 패킷에는 JSON 형태의 데이터 또는 자체적인 바이너리 포맷이 포함된다. 본 연구에서는 프록시 도구를 통해 주요한 응답 값을 반환하는 API를 식별하였으며, Python 3를 통해 주요 API에 대한 네트워크 통신을 재구성하여 데이터를 수집하였다.

제 2 절 분석 방지 우회

드론 조종 어플리케이션에서 일부 어플리케이션 데이터는 암호화가 적용되어 저장되며 네트워크 패킷 재구성 과정에서 어플리케이션별 자체적인 알고리즘으로 요청 패킷에 데이터를 구성하는 경우가 존재한다. 따라서 디지털 증거 수집 과정에서 어플리케이션에 대한 역공학 분석은 필수적이다. 그러나 드론 조종 어플리케이션에는 분석 방지 기능이 존재한다. 본 연구에서는 드론 조종 어플리케이션에 대한 역공학 분석을 위해 분석 방지 기능을 우회하였다.

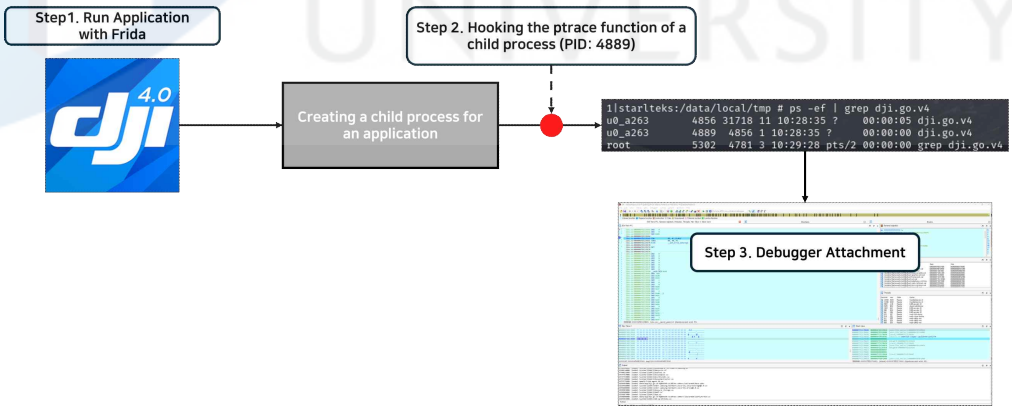
1) 안티 디버깅

안티 디버깅은 프로세스에 디버거가 부착되는 것을 감지·방지하거나 디버깅 관련 유틸리티를 통한 프로세스 접근을 차단하는 기술이다. DJI 드론 조종 어플리케이션에는 총 두 가지의 안티 디버깅 기술이 적용된다.

첫 번째 안티 디버깅 기술은 디버거 부착 방지 기술이며, DJI GO·GO4·Pilot 어플리케이션에 적용되어 있다. 해당 기술은 드론 조종 어플리케이션 실행 시 메인 프로세스 하위에 자식 프로세스를 생성하여, 메인 프로세스를 추적하도록 한다. 자식 프로세스가 메인 프로세스를 추적하는 동안에는 디버거를 메인 프로세스에 부착할 수 없으며, 이는 이미 메인 프로세스가 다른 위치에서 추적되고 있기 때문이다. 이때 자식 프로세스를 임의로 종료 시 메인 프로세스가 함께 종료된다. 본 연구에서는 이러한 디버거 부착 방지 기법을 우회하기 위하여 [그림 2-1]와 같이 Frida를 이용해 DJI 드론 조종 어플리케이션의 실행 초기에 호출되는 fork 함수를 통해 자식 프로세스가 생성되는 시점을 식별하였으며, ptrace 함수를 후킹하여 자식 프로세스의 추적 대상 프로세스를 임의의 프로세스로 변경하였다. 이를 통해 어플리케이션 실행 과정에서 fork 함수로 생성된 자식 프로세스가 수행하는 ptrace 함수의 추적 대상이 메인 프로세스가 아닌, 무관한 프로세스를 가리키도록 설정하였다. 이러한 동작을 수행한 후에는 정상적으로 메인 프로세

스에 디버거가 부착됨을 확인하였다.

두 번째 안티 디버깅 기술은 안티 Frida 기술이며, DJI 드론 조종 어플리케이션 중 DJI Fly에 해당 기술이 적용되어 있다. 안티 Frida는 어플리케이션이 Frida를 통한 프로세스 접근을 감지할 경우 프로세스를 종료하는 기술이다. Frida는 역공학 분석 과정에서 패킹 및 SSL 인증서 고정 기법 우회, 주요 암호화 함수 후킹 등에 활용된다. 이러한 안티 Frida로 인해 Frida를 사용할 수 없을 경우, 어플리케이션에 대한 역공학 분석에 제약이 발생한다. 본 연구에서는 Frida를 통해 DJI Fly 어플리케이션을 실행한 후, dlopen 함수를 후킹하여 특정 라이브러리 로딩 시점과 프로세스 종료 간의 상관관계를 분석하였다. 그 결과, DJI Fly 어플리케이션에서 libmsaoaidsec.so 라이브러리를 로드하는 시점에 프로세스가 종료되는 것을 확인하였다. 이후 해당 라이브러리 내부의 pthread_create 함수를 Frida를 통해 후킹하여 함수 호출 동작을 무력화하였다. 이 과정을 통해 안티 Frida 기능이 우회되었으며, 결과적으로 Frida를 통한 DJI Fly 어플리케이션의 동적 분석이 가능해졌다.



[그림 2-1] DJI 어플리케이션의 디버거 부착 방지 기술 우회 절차

2) 패킹

Android 어플리케이션에서 패킹이란 개발된 어플리케이션의 핵심 코드

와 리소스를 보호하기 위해 암호화, 압축 등을 통해 별도의 래퍼 코드안에 삽입하는 기술을 의미한다. DJI 어플리케이션은 모두 패킹이 적용되어 있어, DEX 파일 또는 so (Shared object) 파일에 대한 역공학 분석 시 주요한 코드를 식별할 수 없다. 본 연구에서는 언패킹된 DEX 파일과 so 파일 획득을 위해 메모리 덤프를 수행하였다. DEX 파일의 경우 어플리케이션이 구동되는 환경에서 Frida를 통해 덤프를 수행하였으며, so 파일 또한 어플리케이션이 구동 중인 시점에서 MemDumper를 실행하여 메모리에서 언패킹된 파일을 수집하였다.

3) SSL 인증서 고정 기법

SSL 인증서 고정 기법은 어플리케이션이 서버와 통신하는 과정에서, 사전에 정의된 특정 인증서만을 신뢰하여 중간자 공격을 방지하는 보안 기술이다. 프록시 도구를 통해 네트워크 통신을 분석하기 위해선 어플리케이션이 동작하는 환경에 프록시 도구의 인증서를 설치하여 프록시 환경을 구성해야 한다. 따라서 어플리케이션에 적용된 SSL 인증서 고정 기법은 이러한 프록시 도구의 인증서를 신뢰하지 않아, 프록시 도구에서 네트워크 통신 내용을 식별할 수 없다. 본 연구에서는 DJI 및 Parrot의 드론 조종 어플리케이션에 적용된 SSL 인증서 고정 기법을 우회하기 위해 Frida를 활용하였다. Frida를 통해 어플리케이션의 인증서 검증 관련 로직을 후킹하여 프록시 도구의 인증서가 탐지되지 않도록 한다. 이러한 접근 방식을 통해 프록시 도구에서 어플리케이션과 서버의 통신 내용을 식별할 수 있게 되었다.

제 3 장 데이터 수집 및 분석 기술

제 1 절 데이터 수집

1) 드론 조종기

드론 조종기에 저장되는 핵심 데이터는 드론 조종 어플리케이션의 앱 데이터이다. 드론 조종기에서는 조종기와 드론을 연결하고, 제어하기 위한 기능을 제공하는 어플리케이션을 구동시킨다. 따라서 조종자의 조종 행위와 연결된 드론 정보는 이러한 드론 조종 어플리케이션을 통해 생성된다.

로컬 환경에 저장된 데이터는 데이터 파티션과 유저 파티션으로 저장 경로가 분류된다. 본 연구에서는 모바일 조종기에서 동작하는 어플리케이션과, DJI SC에서 동작하는 어플리케이션의 저장 경로를 [표 3-1]과 같이 식별하였다. 데이터의 저장 경로는 어플리케이션별로 상이하다.

본 연구의 분석 대상인 모바일 조종기와 DJI SC는 Android 환경이다. 따라서 Android 설정 탭의 개발자 옵션에서 USB 디버깅을 활성화할 경우, USB 포트를 통해 ADB를 통한 접근이 가능하다. 본 연구의 데이터 수집 과정에서는 ADB의 `adb pull` 명령어를 통해 어플리케이션 데이터를 추출하였다. Android 환경에서 ADB를 통한 접근은 일반적으로 유저 파티션 경로에 한정된다. 이는 Android의 권한 구조로 인한 문제이며, 데이터 파티션은 root 권한을 획득한 경우에만 접근할 수 있다. 본 연구에서는 권한을 통한 접근제어를 해결하기 위해 루팅된 모바일 단말기를 대상으로 연구를 수행하였다. DJI SC는 공개된 취약점 연구 결과를 활용하였다. 선행 연구의 취약점은 DJI SC에서 root 권한으로 실행되는 프로세스의 바이너리에 존재하는 OS Command Injection이며, 본 연구에서는 해당 취약점을 활용하여 root 권한으로 데이터 파티션에 저장된 앱 데이터를 유저 파티션으로 복사한 뒤, ADB를 통해 접근하였다.

[표 3-1] ADB를 통한 조종기 데이터 수집 경로

Application	Partition	Data path
DJI GO	Data Partition	/data/data/dji.pilot
	User Partition	/sdcard/DJI/dji.pilot
DJI GO4	Data Partition	/data/data/dji.go.v4
	User Partition	/sdcard/DJI/dji.go.v4
DJI Fly	Data Partition	/data/data/dji.go.v5
	User Partition	/sdcard/Android/data/dji.go.v5
DJI Pilot	Data Partition	/data/data/com.dji.industry.pilot
	User Partition	/sdcard/Android/data/ com.dji.industry.pilot
DJI SC	Data Partition	/data/data/dji.go.v4 /data/data/dji.go.v5 /data/data/com.dji.industry.pilot
	User Partition	/sdcard/DJI/dji.go.v4 /sdcard/Android/data/dji.go.v5 /sdcard/Android/data/com.dji.industry.pilot
Parrot Freeflight 6	Data Partition	/data/data/com.parrot.freeflight6
	User Partition	/sdcard/DCIM/PI*

2) 드론 클라우드

드론 조종 어플리케이션은 클라우드 서버에 데이터를 동기화한다. 클라우드에 동기화되는 데이터는 대표적으로 어플리케이션에 로그인 한 조종자의 프로필 정보, 비행로그 파일 등이 있다. 이러한 클라우드 데이터는 조종자가 드론 조종 어플리케이션을 조종기에서 삭제하거나, 조종기의 로컬 데이터를 초기화한 경우에도 서버 측에 유지되며, 사용자의 크리덴셜을 확보

한 상황에서 특정 조종기에 종속되지 않고 접근할 수 있다. 이러한 드론 조종 어플리케이션은 클라우드 서버의 API와의 통신을 통해 데이터를 송·수신 한다. 본 연구에서는 드론 조종 어플리케이션이 클라우드 서버에 전송하는 네트워크 패킷을 재구성하여, 클라우드 서버로부터 데이터를 수집한다. 연구 대상인 DJI와 Parrot의 드론 조종 어플리케이션은 클라우드 서버와의 통신 과정에서 HTTP/HTTPS 프로토콜을 활용한다. 따라서 올바른 응답 값을 획득하기 위해 API별 HTTP Header, Body, Parameter, Method를 구성하였으며, 일부 API에 존재하는 보안 기능을 우회하였다. 본 연구에서 식별한 DJI 클라우드 서버의 주요 API는 [표 3-2]와 같다.

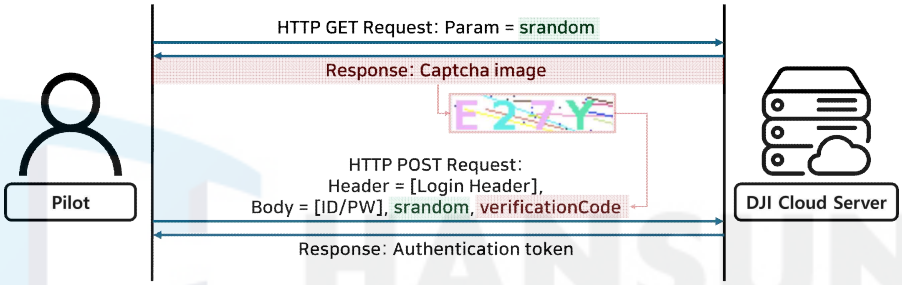
[표 3-2] DJI 클라우드의 주요 API 식별 결과

API number	Purpose	API	Request component
API-1	Get Captcha Image	https://account-api.dji.com/apis/apprest/v1/vcode	GET, Param: {srandom:[srandom string]}
API-2	Get Token	https://account-api.dji.com/apis/apprest/v1/email_login	POST, Header: { Content-Type, Accept-Encoding, InvokeId-Mc, Sign-Mc, DeviceId-Mc, AppId-Mc, ClientName-Mc, Timestamp-Mc, Encrypted-Mc} Body:{email:[DJI ID], password:[DJI PW], srandom:[srandom string],verificationCode: [verification code]}
API-3	Get Pilot Information	https://www.skypixel.com/api/profiles/my	GET, Param: {token:[token]}

API-4	Get Connected Drone Information	https://mydjiflight.dji.com/flight/query/userinfo	GET, Param: {token:[token]}
API-5	Get Flight Record List	https://mydjiflight.dji.com/flight/list	GET, Param: {token:[token], page:1, page_size:99}
API-6	Get Flight Record List	https://mydjiflight.dji.com/flight/record_info_list	GET, Param: {token:[token], page:1, page_size:99}
API-7	Get Flight Record Download	https://mydjiflight.dji.com/flight/zipdownload	GET, Param:{token:[token], filename:[TXT Flight Record filename]}
API-8	Convert S/N	https://djigo-hk.djiservice.org/api/v1/djicare/sn_product_center/query	GET, Param:{sn:[S/N string]}
API-9	Get Flight History Information	https://mydjiflight.dji.com/flight/aircraft_info	GET, Headers:{app-id:dji-fly, signature:[signature]}, Param:{sn:[Drone's S/N]}
API-10	Get Pilot Binding Information	https://api.djiservice.org/api/device-manager/care/bind-sn-list	POST, Headers:{app-id:dji-fly, signature:[signature]}, Body:{token:[token]}
API-11	Get Drone Binding Information	https://api.djiservice.org/api/device-manager/care/device-detail	GET, Headers:{app-id:dji-fly, signature:[signature]}, Param:{sn:[Drone's S/N]}

API-1, 2는 로그인을 위해 활용된다. [그림 3-1]은 DJI 클라우드 서버를 통한 로그인 API 호출 절차의 도식화 결과이다. 조종자는 API-1에 대한 요청 패킷을 구성하는 과정에서 HTTP Parameter에 srandom 값을 구성하여 HTTP GET 메서드를 통해 클라우드 서버로 전송하고, 캡차 이

미지를 수신한다. 이후 API-2에 대한 요청 패킷을 구성하는 단계에서 캡차 이미지의 4자리 문구와 srandom, ID/PW를 HTTP Body에, HTTP Header에는 Login Header를 구성하여 API-2에 전송 후, 조종자의 인증 토큰을 획득한다. HTTP Body에 포함되는 DJI 어플리케이션의 PW는 암호화된 상태로 요청 패킷의 password 필드에 추가된다. [수식 3-1]은 password 필드의 암호화 알고리즘이다. DJI 어플리케이션의 PW를 AES128-CBC-PKCS5Padding을 통해 암호화한다. 이때 암호 키와 IV (Initialization vector)는 고정된 값을 사용한다.



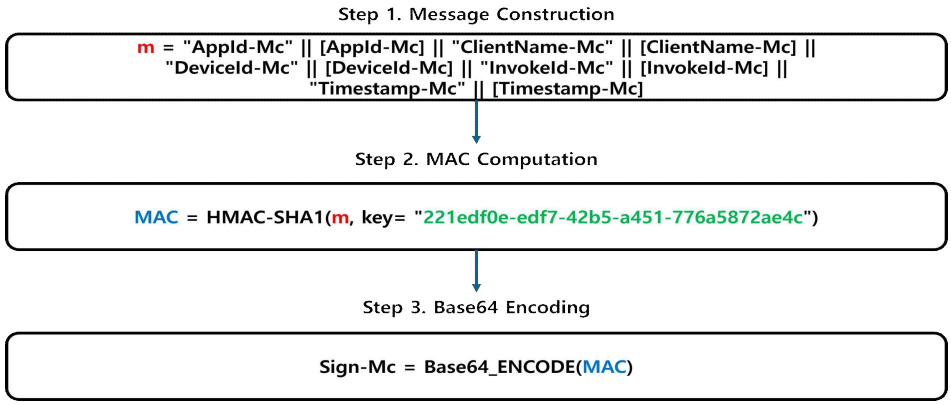
[그림 3-1] DJI Cloud 서버를 통한 로그인 API 호출 절차

$$C = AES128 - CBC - PKCS5Padding_{Encrypt}(P, Key, IV)$$
$$P = DJI\ password$$
$$Key = 221edf0e - edf7 - 42$$
$$IV = 1wdv\$ESZ@QAX3rfc$$

[수식 3-1] API-2의 password 필드 암호화 알고리즘

HTTP Header는 총 7개의 Sign 값과 2개의 고정된 값으로 구성된다. 본 연구에서는 이러한 Header 구성 요소를 역공학 분석을 통해 식별하였다. HTTP Header에서 Accept-Encoding 필드는 'identity'를, Content-Type 필드는 'application/x-www-form-urlencoded'로 설정한다. Encrypted-Mc와 AppId-Mc 필드는 각각 '1'과 'djigo'가 구성된다. InvokeId-Mc와 DeviceId-Mc는 모두 UUID 형식이며, 두 필드는 모두 임의의 값으로 설정하더라도 클라우드 서버는 유효한 값으로 인식한다. 그러나, DeviceId-Mc는 동일한 값을 반복적으로 사용할 수 있는 반면,

InvokeId-Mc는 매 요청마다 이전 요청과는 다른 UUID 값을 구성해야 한다. ClientName-Mc 필드는 DJI 어플리케이션에서 기본적으로 'android-9-4.3.61'과 같은 형식으로 Android 버전 정보를 담고 있지만, 패킷 재구성 과정에서 빈 값이 아닌 임의의 값을 구성하여도 정상적인 값으로 인식된다. Timestamp-Mc 필드는 유닉스 타임스탬프 형식의 시간 정보를 가지며, DJI 어플리케이션에서는 요청 전송 시점의 시간 값을 갖지만, 패킷 재구성 과정에서 임의의 고정된 시간 값으로 구성하여도 정상적인 값으로 인식된다. Sign-Mc는 HTTP Header의 5개의 필드를 기반으로 생성되는 MAC (Message Authentication Code) 값이다. [그림 3-2]와 같이 해당 값은 메시지 m 생성, MAC 계산, Base64 인코딩의 단계를 거쳐 생성된다. 메시지 m은 AppId-Mc, ClientName-Mc, DeviceId-Mc, InvokeId-Mc, Timestamp-Mc 순서로, 각 필드명과 그에 대응하는 값을 결합한 형태의 문자열이다. 이후 메시지 m을 고정된 키와 HMAC-SHA1 알고리즘을 통해 MAC 값을 생성한 후, Base64로 인코딩하여 최종 Sign-Mc를 생성한다. 이러한 값을 통해 API-2에 대한 요청 패킷을 구성 후 HTTP POST 메서드를 통해 클라우드 서버로 전송할 경우, 정상 패킷으로 인식된다. API-2에서는 InvokeId-Mc와 Sign-Mc를 통해 단순 변조와 재구성이 불가능하며, 본 연구에서는 역공학 분석을 통해 식별한 알고리즘을 기반으로 API-2에 대한 임의 패킷 재구성에 성공하였다.



[그림 3-2] Sign-Mc 생성 절차

API-3, 4, 5, 6, 7, 8은 HTTP Parameter에 인증 토큰과 각 API에서 요구하는 필드에 값을 구성하여 패킷을 구성한 뒤, HTTP GET 메서드를 통해 클라우드 서버로 전송할 수 있다.

API-9, 10, 11은 앞선 다른 API들과 같이 HTTP Header, Body, Parameter에 각 API에서 요구하는 필드 값을 구성해야 한다. 이때 HTTP Header에 signature 필드를 요구한다. DJI 클라우드 서버에서 signature는 HTTP 요청 패킷의 일부 필드를 기반으로 생성된다. [수식 3-2]는 API-9에서 요구하는 signature 필드 값을 생성하는 알고리즘이다. 요청 패킷에 포함되는 드론의 S/N과 고정된 키 값을 기반으로 HMAC-SHA256 알고리즘을 통해 생성된 값이 signature 필드의 값이다.

$$signature = HMAC-SHA256(m, key)$$
$$key = 59725a6a5a4a6c5442777a747046334d4f6d76464b53627131304e4959456f39$$
$$m = Drone's\ S/N$$

[수식 3-2] API-9의 signature 생성 알고리즘

[수식 3-3]은 API-10과 11에 대한 signature 필드 값을 생성하는 알고리즘이다. API-10은 조종자의 인증 토큰을, API-11은 드론의 S/N를 m으로 사용한다. 이후 고정된 키와 HMAC-SHA256 알고리즘을 통해 signature를 생성한다. 이는 앞선 [수식 3-2]와 방법은 동일하지만, 고정된 키 값에서 차이가 존재한다. 본 연구에서는 이러한 signature 생성 알고리즘을 식별하였으며, 이를 기반으로 API-9, 10, 11에 대한 임의의 요청 패킷을 구성하였다.

$$signature = HMAC-SHA256(m, key)$$
$$key = 78706b6459635a73377a547733554d54434b334d624e6e6e377178477566484445$$
$$m = Drone's\ S/N \,||\, Pilot's\ Token$$

[수식 3-3] API-10, 11의 signature 생성 알고리즘

본 연구에서 식별한 Parrot 클라우드 서버의 주요 API는 [표 3-3]와 같다. DJI 클라우드 API와 동일하게 HTTP/HTTPS 프로토콜을 활용하며, HTTP

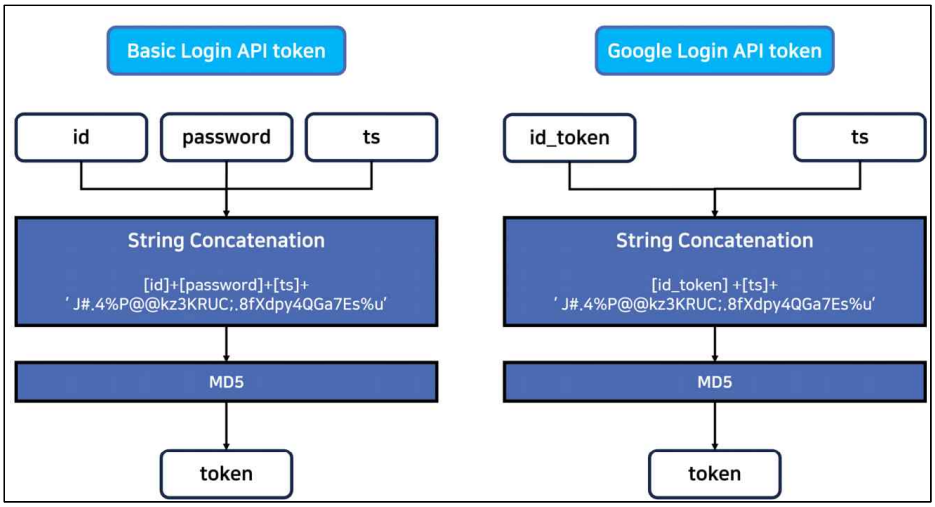
Header, Body, Parameter에 적절한 값을 구성하여 패킷을 재구성할 수 있다.

[표 3-3] Parrot 클라우드의 주요 API 식별 결과

API number	Purpose	API	Request component
API-12	Get Token	https://accounts.parrot.com /V4/user/auth	POST, Header:{ x-callerid:FF6_Android} Body:{ id=[Parrot ID], password=[Parrot PW]} Parameter:{ token, ts}
API-13	Get Token	https://accounts.parrot.com /V4/user/fromsn/google	POST, Header:{ x-callerid:FF6_Android} Body:{ id=[Parrot ID], password=[Parrot PW]} Parameter:{ token, ts}
API-14	Get Pilot Information	https://droneacademy.parrot .com/api4/profile	GET, Header:{ authentication:[Token]}
API-15	Get Flight record file	https://droneacademy.parrot .com/api4/runs	GET, Header:{ authentication:[Token]} Parameter:{ start_date, end_date, product_id}
API-16	Get Flight record file	https://droneacademy.parrot .com/api4/runs/[id]/details	GET, Header:{ authentication:[Token]}

API-12과 13은 로그인 기능을 가진 API로 응답 값으로 HTTP POST 요청 시, 조종자의 인증 토큰을 반환한다. API-12은 Parrot 클라우드의

일반 로그인 기능으로, 로그인 시 Parrot 어플리케이션의 크리덴셜만을 요구한다. 반면 API-13는 구글 OAuth를 통한 로그인으로, 로그인 시 Parrot 어플리케이션 크리덴셜이 아닌 로컬 아티팩트에 존재하는 Oauth 토큰인 id_token을 요구한다. 이는 조종자가 Parrot Freeflight 6 어플리케이션에서 구글을 통한 로그인 시 로컬아티팩트에 생성된다. id_token에 대한 로컬 아티팩트 분석 결과는 2절에서 자세히 설명한다. 이러한 로그인 API는 Header에 고정된 값인 "x-callerid:FF6_Android"가 구성되며, HTTP Body에는 Parrot 어플리케이션의 ID/PW 정보가 구성된다. ts는 패킷 전송 시각에 대한 유닉스 타임스탬프이며, token은 ts와 조종자의 크리덴셜 값으로 생성된 값이다. 본 연구에서는 Parrot Freeflight 6 어플리케이션에 대한 역공학 분석을 통해 [그림 3-3]과 같이 각 로그인 API별 token 생성 알고리즘을 식별하였다. 기본 로그인 API의 token은 Parrot 어플리케이션 크리덴셜인 ID/PW와 API 요청 시 구성한 ts를 활용한다. 세 가지 값을 '[id]+[password]+[ts]' 형태로 연접한 뒤, 하위에 고정 문자열인 'J#.4%P@@kz3KRUC;.8fXdpy4QGa7Es%u'를 추가한다. 이후 연접 문자열을 MD5를 통해 해시 값을 생성하며, 이를 token으로 활용한다. 구글 로그인 API의 token은 앞선 기본 로그인 API의 token 생성 과정과 동일하며, 입력 값이 id_token과 ts로 구성된다는 차이점이 존재한다.



[그림 3-3] Parrot Cloud의 로그인 API token 생성 알고리즘 도식화

API-14, 15, 16은 각각 조종자의 정보와 비행로그 파일을 획득하기 위한 API이다. 이러한 API는 HTTP GET 요청을 통해 패킷을 전송할 수 있으며, DJI 클라우드 API와 마찬가지로, HTTP Header, Body, Parameter에 적절한 값을 구성해야 한다.

앞서 서술한 바와 같이, DJI와 Parrot의 클라우드 서버의 API에 대해 네트워크 패킷을 재구성할 수 있다. 이러한 네트워크 요청의 응답 값은 주로 JSON 또는 HTTP Body에 파일이 구성된 형태이다. 본 연구에서는 네트워크 응답 패킷을 분석하여, 디지털 포렌식 관점에서 의미있는 데이터를 획득하였다.

제 2 절 드론 조종기 데이터 분석

본 연구에서 드론 조종 어플리케이션 데이터를 분석한 결과는 [표 3-4]와 같으며, 분석 결과에 따라 총 5가지 유형인 드론 기체 정보, 조종자 정보, 비행로그 파일, 미디어 파일, 어플리케이션 로그로 분류하였다.

[표 3-4] 드론 조종 어플리케이션 데이터 분석 결과

Application	Partition	File name	Acquirable Data	Encrypted
DJI GO	/data/data/dji.pilot	files/user_avatar.png	User profile image	×
		databases/dji.db	Drone serial number, firmware file information	×
	/sdcard/DJI/dji.pilot	DJI Album/org_[16 character string]_[13 digit number].jpg	Captured photos, capture dates	×
		DJI Album/org_[16 character string]_[13 digit number].mov	Captured videos, capture dates	×
		FlightRecord/SyncResult\DJIFRSyncLog_[yyyy]-[mm]-[dd]_[hh-mm-ss].txt	Pilot email, synchronization logs	×
		FlightRecord/.Overview-[Email].tmp	Pilot email, Pilot's drone flight information	×
		FlightRecord/userinfo.	Pilot's drone flight information	×

		json		
		FlightRecord/ DJIFlightRecord_[yyyy]-[mm]-[dd]_[hh-mm-ss].txt	Drone flight information	○
		DJI_RECORD/ [yyyy]_[mm]_[dd]_[hh]_[mm]_[ss].mp4	Captured video files and capture dates	×
		DJI_RECORD/ [yyyy]_[mm]_[dd]_[hh]_[mm]_[ss].info	Capture dates of recorded videos	×
		Upgrade/DLPackage/ [Drone model name]_FW_V[Version].bin	Drone firmware files	×
		CACHE_IMAGE/ thumb_[16 character string]_[13 digit number].jpg	Thumbnail files of recorded videos	×
		CACHE_IMAGE/ screen_[16 character string]_[13 digit number].jpg	Cache files of captured photos	×
		CACHE_IMAGE/ screen_[16 character string]_[13 digit number].info	Capture dates of photos	×
DJI GO4	/data/data/ dji.go.v4	databases/dji.db	Drone serial number	×
		files/user_avatar.png	Pilot's profile image	×
		files/mmkv/ mmkv.default	Drone serial number, Pilot's authentication token	×
	/sdcard/ DJI/dji.go.v4	CACHE_IMAGE/ thumb_[string]_[number].jpg	Thumbnail image files of recorded videos	×
		DJI_RECORD/ [yyyy]_[mm]_[dd]_[hh]_[mm]_[ss].info	Recording date, video duration, location information, video resolution of recorded videos	×

		DJI_RECORD/ [yyyy]_[mm]_[dd]_[hh]_ [mm]_[ss].mp4	Recorded video files	×
		FlightRecord/ DJIFlightRecord_[yyyy]-[mm]-[dd]_ [hh-mm-ss].txt	Drone flight information	○
		LOG/ERROR_POP/LOG [dd]-[mm]-[yyyy]-[Drone serial number]	In-flight drone events	×
DJI Fly	/data/data/dji.go.v5	files/user_avatar.png	Pilot's profile image	×
		files/mmkv/ mmkv.default	Drone/Drone's battery/Drone controller serial number, Pilot's authentication token	×
		/databases/dji.db	Drone serial number	x
		/databases/flysafe_license_unlock.db	Pilot's authentication token	x
	/sdcard/Android/data/dji.go.v5	/files/DJI_Temp/tmp_ avatar.png	Pilot's profile image	x
		/files/FlightRecord/DJIFlightRecord_[yyyy-mm-dd]_[hh-mm-ss].txt	TXT Flight Record File	O
		/files/FlightRecord/MCDatFlightRecords/[yyyy-mm-dd]_[hh-mm-ss]_ FLY[Count].dat	DAT Flight Record File	O
		/files/MediaCaches/ [yyyy_mm_dd_hh_mm_ss]_Cache.mp4	Cached recorded video files	x
		/files/FD_h264cache/ [Drone serial number]	A directory named after the connected drone's serial number	x
		/files/FD_h264cache//[Drone serial number]/thumbnail.jpeg	Image files captured during drone boot	x
		/files/LOG/CACHE/	Cached Pilot authentication	o

		AccountCenter/log-[yyyy]-[mm]-[dd].log	tokens	
		/files/LOG/CACHE/deviceManager/log-[yyyy]-[mm]-[dd].log	Cached serial numbers and binding information of the drone and RC controller	o
		/files/LOG/CACHE/AeroScopeLogic/log-[yyyy]-[mm]-[dd].log	Cached latitude and longitude of the DJI RC	o
		/cache/ImageCaches/Photo_[number]_[media file name]_[number]_0_[Capture timestamp]_photo_thumbnail.jpg	Thumbnail files of captured photos	x
		/cache/ImageCaches/Video_[number]_[media file name]_[number]_0_[Capture timestamp]_photo_thumbnail.jpg	Thumbnail files of recorded videos	x
	/sdcard//DCIM/DJI Album	/dji_fly_YYYYMMDD_HHMMSS_Number_UnixTime(millisecond)_photo.jpg	Photos captured by the drone	x
		/dji_fly_YYYYMMDD_HHMMSS_Number_UnixTime(millisecond)_video.mp4	Video files recorded by the drone	x
DJI Pilot	/data/data/com.dji.industry.pilot	files/mapsnap/[32 character string].png	Thumbnail files for pilot-defined “flight missions”	x
		databases/flysafe_license_unlock.db	Pilot authentication tokens	x
	/sdcard/Android/data/com.dji.indust	/files/DJI/com.dji.industry.pilot/CACHE_IMAGE/thumb_[16 character string]_[13	Thumbnail images of photos and videos captured by the drone	x

	ry.pilot	digit number].jpg		
		/files/DJI/com.dji.industry.pilot/FlightRecord/DJIFlightRecord_[yyyy-mm-dd]_[hh-mm-ss].txt	TXT Flight Record file	x
	/sdcard/DJI/com.dji.industry.pilot/Pilot Album	org_[16 character string]_[13 digit number].jpg	Photos captured by the drone	x
		org_[16 character string]_[13 digit number].mov	Videos recorded by the drone	o
DJI SC(DJI GO4)	/data/data/dji.go.v4	files/mmkv/mmkv.default	Drone serial number and pilot authentication token	x
		files/user_avatar.png	Pilot's profile image	x
		databases/flysafe_license_unlock.db	Pilot's authentication token	x
		databases/dji.db	Drone serial number	x
	/sdcard/DJI/dji.go.v4	DJI_RECORD/[yyyy_mm_dd_hh_mm_ss].mp4	Video files recorded by the drone	x
		DJI_RECORD/[yyyy_mm_dd_hh_mm_ss].info	Recording date/time and location information for video files	x
		CACHE_IMAGE/pov_*	Cached image files	x
		CACHE_IMAGE/thumb_[16 character string]_[Unix timestamp].jpg	Cached image files	x
		FlightRecord/DJIFlightRecord_[yyyy-mm-dd]_[hh-mm-ss].txt	TXT Flight Record File	o
		FlightRecord/MCDatFlightRecords/[yy-mm-dd-hh-mm-ss_FLY[COUNT].DAT	DAT Flight Record File	o
		FlightRecord/userinfo.json	Drone flight information per drone operated by the pilot	x
		LOG/ERROR_POP_LOG/	In-flight drone events	o

		[dd]-[mm]-[yyyy]-[Drone serial number]		
DJI SC (DJI Fly)	/data/data/dji.go.v5	files/mmkv/mmkv.default	Drone and controller serial numbers	x
		files/user_avatar.png	Pilot's profile image	x
		databases/dji.db	Drone serial number	x
		databases/flysafe_license_unlock.db	Pilot's authentication token	x
	/sdcard/DJI/dji.go.v5	files/LOG/CACHE/deviceManager/log-[yyyy]-[mm]-[dd].log	Serial numbers and binding information of the drone and RC controller	o
		files/LOG/CACHE/AeroScopeLogic/log-[yyyy]-[mm]-[dd].log	Latitude and longitude of the DJI RC	o
		FlightRecord/DJIFlightRecord_[yyyy-mm-dd]_[hh-mm-ss].txt	TXT Flight Record File	o
		FlightRecord/MCDatFlightRecords/[yy-mm-dd-hh-mm-ss_FLY[COUNT].DAT	DAT Flight Record File	o
		cache/ImageCaches/Photo_[number]_[media file name]_[number]_0_[Capture timestamp]_photo_thumbnail.jpg	Thumbnail files of captured photos	x
		cache/ImageCaches/Video_[number]_[media file name]_[number]_0_[Capture timestamp]_photo_thumbnail.jpg	Thumbnail files of recorded videos	x
		DJI FLY/Video/OriginalFiles/Photo/*.jpg	JPG files captured by drones	x
		DJI FLY/Video/*.mp4	MP4 files captured by drones	x

DJI SC (DJI Pilot)	/data/data/com.dji.industry.pilot	flysafe_license_unlock.db	Pilot's authentication token	x
	/sdcard/DJI/com.dji.industry.pilot	FlightRecord/DJIFlightRecord_[yyyy-mm-dd]_[hh-mm-ss].txt	TXT Flight Record File	o
		FlightRecord/MCDatFlightRecords/[yy-mm-dd-hh-mm-ss_FLY[COUNT].DAT	DAT Flight Record File	o
		CACHE_IMAGE	Cached thumbnail images	x
		Pilot Album/org_[len 16 string]_[unix time stamp].jpg	JPG files captured by drones	x
		Pilot Album/org_[len 16 string]_[unix time stamp].mp4	MP4 files captured by drones	x
Parrot Freeflight6	/data/data/com.parrot.freeflight6	files/ACADEMY/flightDetailes/[len 4 num]_[yyyy]-[mm]-[dd]T[hhmmss]+0900_[len 6 string]_[fly count].json	Flight Record Detail File	x
		files/ACADEMY/flightSummary/[len 4 num]_[yyyy]-[mm]-[dd]T[hhmmss]+0900_[len 6 string]_[fly count].json	Flight Record Summary File	x
		shared_prefs/account.xml	Last login time (Unix timestamp)	x
		shared_prefs/activation.xml	Serial numbers of the connected drone and RC controller	x
		shared_prefs/ff6Prefs.xml	Serial numbers of the most recently connected RC controller and drone	x
		shared_prefs/com.googl	Google OAuth authentication	x

		e.android.gms.signin.xml	token and email address	
	/sdcard/ DCIM	[Drone model name]/ [Drone serial number]/ single-photo/*JPG	Photo files captured by the drone	x
		[Drone model name]/ [Drone serial number]/ video/*MP4	Video files recorded by the drone	x

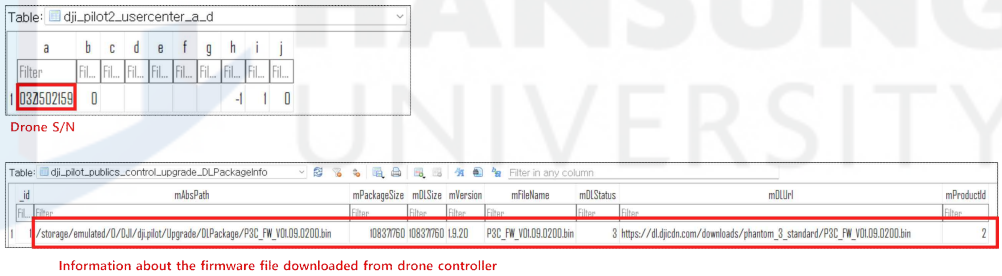
1) 드론 기체 정보

분석된 데이터에서 식별된 드론 기체 정보는 드론 시리얼 번호와 펌웨어 파일이며, 이러한 데이터는 드론 조종기에 연결된 드론을 추적하는 데 활용할 수 있다. 각 드론 조종 어플리케이션별 드론 기체 정보 식별 결과는 [표 3-5]와 같다.

[표 3-5] 드론 기체 관련 데이터 분류

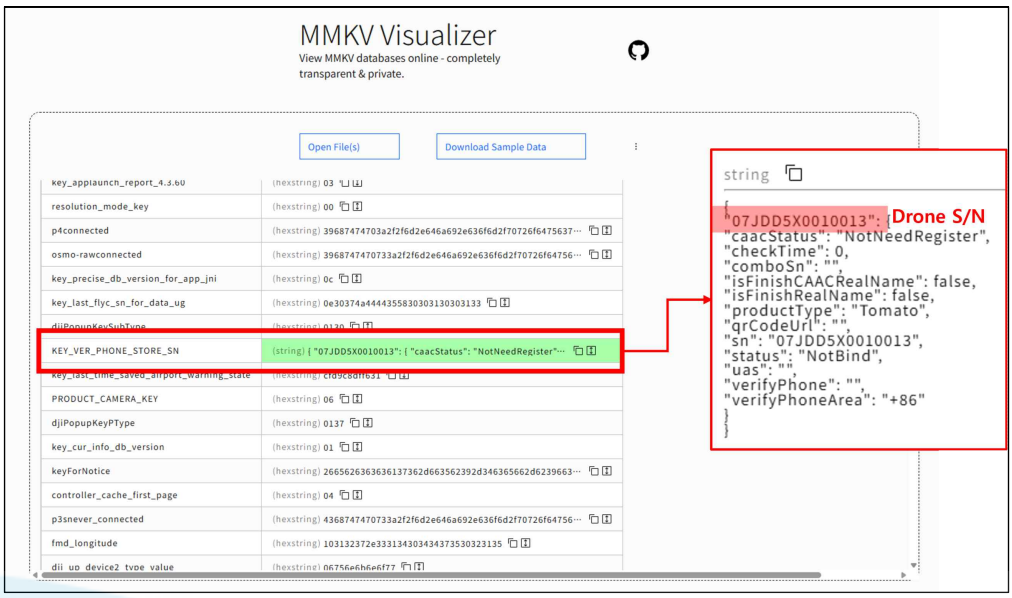
Application	File name	Data
DJI GO	dji.db	Drone S/N, Firmware information
	[Drone model name]_ FW_V[Version].bin	Drone firmware file
	userinfo.json	Drone S/N
DJI GO4	dji.db	Drone S/N
	mmkv.default	Drone S/N
DJI Fly	dji.db	Drone S/N
	mmkv.default	Drone S/N
	FD_h264cache/ [Drone serial number]	Drone S/N
DJI SC (DJI GO4)	mmkv.default	Drone S/N
	dji.db	Drone S/N
DJI SC (DJI Fly)	mmkv.default	Drone S/N
	dji.db	Drone S/N
Parrot Freeflight 6	activation.xml	Drone S/N
	ff6Prefs.xml	Last connected drone S/N

DJI GO의 dji.db 파일은 '/data/data/dji.pilot/databases' 경로에 저장된 데이터베이스 유형의 파일이다. [그림 3-4]와 같이 해당 데이터베이스 파일의 'dji_pilot2_usercenter_a_d' 테이블 내 a 컬럼에는 DJI GO 어플리케이션을 통해 비행한 드론의 시리얼 번호가 기록된다. 또한 'dji_pilot_publics_control_upgrade_DLPackageInfo' 테이블에는 모바일 조종기로 다운로드한 펌웨어 파일 정보가 기록된다. 해당 테이블에서 mAbsPath 컬럼에는 펌웨어 파일이 조종기의 로컬 저장소에 저장된 경로를, mfileName 컬럼은 펌웨어 파일명을 의미하며, mDUrl 컬럼에는 펌웨어 파일을 다운로드한 주소가 저장된다. 조종기의 로컬 저장소에는 mAbsPath 컬럼에서 명시된 경로에 실제 'bin' 확장자의 펌웨어 파일이 저장된다. '/sdcard/DJI/dji.pilot/FlightRecord' 경로에는 JSON 파일 유형의 userinfo.json 파일이 존재하며, 조종자가 연결하여 비행한 드론의 시리얼 번호가 기록된다.



[그림 3-4] DJI GO의 dji.db 분석을 통한 드론 정보 식별 결과

DJI GO4의 '/data/data/dji.go.v4/databases' 경로에 존재하는 dji.db에 DJI GO4 어플리케이션을 통해 조종한 드론의 시리얼 번호가 저장된다. 해당 데이터는 dji.db의 'dji_pilot2_usercenter_control_DJICareModel' 테이블 내 mc 컬럼에 저장된다. '/data/data/dji.go.v4/files/mmkv' 경로에 존재하는 mmkv.default 파일은 DJI GO4 어플리케이션에 연결한 드론의 시리얼 번호를 저장한다. 이러한 데이터는 해당 파일의 Key-Value 형태의 구조에서 Key 값이 KEY_VER_PHONE_STORE_SN인 Value에서 식별할 수 있다.



[그림 3-5] DJI GO4의 mmkv.default 분석을 통한 드론 정보 식별 결과

DJI Fly의 '/data/data/dji.go.v5/databases' 경로에는 데이터베이스 유형의 파일인 dji.db가 저장된다. 해당 파일의 'dji_component_activate_model_DJICareModelV2' 테이블 내 mc 컬럼에는 DJI Fly를 통해 조종한 드론의 시리얼 번호가 저장된다. '/data/data/dji.go.v5/files/mmkv' 경로에 저장되는 mmkv.default 파일에도 드론의 시리얼 번호가 저장되며, Key 값이 'key_last_flyc_sn_for_data_ug'인 Value에서 식별할 수 있다. '/sdcard/Android/data/dji.go.v5/files/FD_h264cache' 경로에는 DJI Fly를 통해 조종한 드론의 시리얼 번호를 이름으로 갖는 디렉터리가 존재한다. 따라서 이러한 디렉터리의 존재 여부를 통해 조종된 드론의 시리얼 번호를 식별할 수 있다.

DJI SC의 DJI GO4·Fly 데이터 저장 경로에는 dji.db 파일이 저장된다. 해당 파일은 어플리케이션별로 '/data/data/dji.go.v4/databases'와 '/data/data/dji.go.v5/databases' 경로에 저장되며, 내부에 드론 시리얼 번호가 저장된다. 이때 드론 시리얼 번호가 저장되는 데이터베이스의 테이블과 컬럼은 모바일 조종기의 DJI GO4·Fly의 dji.db 파일과 동일하다. 추

가로 mmkv.default 파일 또한 DJI SC의 DJI GO4·Fly의 '/data/data/[package name]/files/mmkv' 경로에 존재한다. 두 어플리케이션의 mmkv.default 파일 모두 Key가 'key_last_flyc_sn_for_data_ug'인 Value에 드론의 시리얼 번호가 저장된다.

Parrot Freeflight 6의 '/data/data/com.parrot.freeflight6/shared_prefs' 경로의 activation.xml 파일에는 [그림 3-6]와 같이 연결한 드론의 시리얼 번호가 저장된다. 해당 파일에는 드론과 조종기의 시리얼 번호가 모두 기록되며, 선행적으로 기록된 데이터가 드론의 시리얼 번호를 의미한다. 또한 activation.xml와 동일한 경로에 존재하는 ff6Prefs.xml 파일에는 마지막으로 연결한 드론의 시리얼 번호가 저장된다. [그림 3-7]와 같이 드론 시리얼 번호는 Key 값이 'last_drone_uid'인 Value에 기록된다.

```
1  <?xml version='1.0' encoding='utf-8' standalone='yes' ?>
2  <map>
3      <set name="devices">
4          <string>PI040445AC0G005359</string>   Drone S/N
5          <string>PI040443DA9J163046</string>
6      </set>
7  </map>
```

[그림 3-6] Parrot Freeflight 6의 activation.xml 분석을 통한 드론 정보 식별 결과

```
1  <?xml version='1.0' encoding='utf-8' standalone='yes' ?>
2  <map>
3      <boolean name="map_preload_tuto_displayed" value="true" />
4      <string name="map_preload_camera_position">{&amp;quot;bearing&amp;quot;:0
5      <string name="last_remote_control_uid">PI040443DA9J163046</string
6      <boolean name="is_onboarding_finished" value="true" />
7      <int name="skycontroller_model_key" value="0" />
8      <int name="phone_flight_count" value="0" />
9      <boolean name="user_got_hand_launch_mode" value="true" />   Drone S/N
10     <string name="last_drone_uid">PI040445AC0G005359</string>
11     <boolean name="is_tou_accepted" value="true" />
12 </map>
```

[그림 3-7] Parrot Freeflight 6의 ff6Prefs.xml 분석을 통한 드론 정보 식별 결과

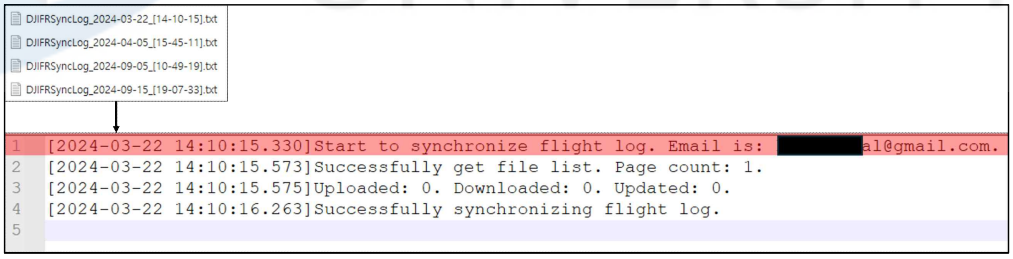
2) 조종자 정보

드론 어플리케이션 데이터 중 조종자와 관련된 데이터는 어플리케이션 프로필 정보, 인증 토큰, 조종기 시리얼 번호가 해당된다. [표 3-6]는 본 연구에서 식별한 조종자 관련 데이터의 분류이다.

[표 3-6] 드론 조종자 관련 데이터 분류

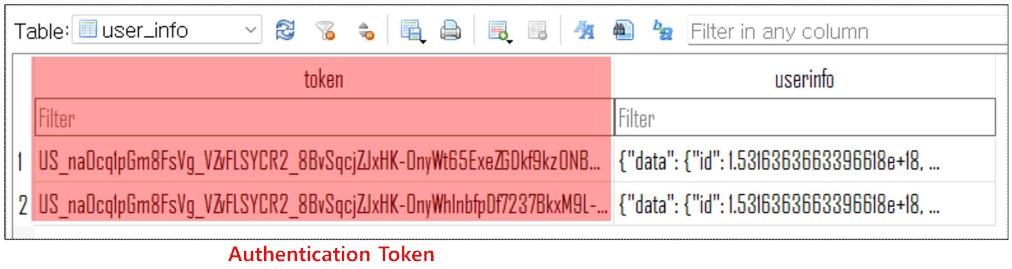
Application	File name	Data
DJI GO	user_avatar.png	Pilot's profile image
	DJIFRSyncLog_[yyyy]-[mm]-[dd]_[hh-mm-ss].txt	Pilot email
	userinfo.json	Pilot email
	.Overview-[Email].tmp	Pilot email
DJI GO4	flysafe_license_unlock.db	Pilot authentication token
	user_avatar.png	Pilot's profile image
	mmkv.default	Pilot authentication token
DJI Fly	user_avatar.png	Pilot's profile image
	flysafe_license_unlock.db	Pilot's authentication token
	tmp_avatar.png	Pilot's profile image
	mmkv.default	Pilot authentication token, Drone controller serial number
DJI Pilot	flysafe_license_unlock.db	Pilot authentication token
DJI SC (DJI GO4)	mmkv.default	Pilot email
	user_avatar.png	Pilot's profile image
	flysafe_license_unlock.db	Pilot authentication token
DJI SC (DJI Fly)	mmkv.default	Drone controller serial number
	user_avatar.png	Pilot's profile image
	flysafe_license_unlock.db	Pilot authentication token
DJI SC (DJI Pilot)	flysafe_license_unlock.db	Pilot authentication token
Parrot Freeflight 6	account.xml	Last login time, Pilot nickname
	activation.xml	Drone controller serial number
	ff6Prefs.xml	Serial number of the most recently connected controller
	com.google.android.gms.signin.xml	Google OAuth token, email address

DJI GO의 '/data/data/dji.pilot/files' 경로에는 조종자의 어플리케이션 프로필 이미지가 PNG 형태인 user_avatar.png 파일에 저장된다. 해당 파일은 DJI GO 뿐만 아니라, DJI GO4·Fly·SC의 데이터에서도 동일하게 존재하며, DJI Fly에서는 'tmp_avatar.png' 파일이 추가적으로 존재한다. '/sdcard/DJI/dji.pilot/FlightRecord/SyncResult' 경로에는 'DJIFRSyncLog_[yyyy]-[mm]-[dd]_[hh-mm-ss].txt' 파일이 존재한다. 해당 파일은 [그림 3-8]와 같이 조종자가 클라우드에 비행로그 파일 동기화 행위를 수행한 로그가 기록되는 파일이다. 해당 파일에 기록되는 로그에는 동기화 로그 뿐만 아니라, 동기화를 수행한 조종자의 이메일 정보가 존재한다. '/sdcard/DJI/dji.pilot/FlightRecord' 경로에는 조종자의 이메일이 저장된 userinfo.json 파일이 존재한다. 해당 파일은 JSON 형태의 파일이며, Key가 'Email'인 Value에 조종자의 이메일이 평문으로 저장된다. 또한 '/sdcard/DJI/dji.pilot/FlightRecord' 경로에는 파일명에 조종자의 이메일이 기록되는 '.Overview-[Email].tmp' 파일이 저장된다. 해당 파일은 DJI GO를 통해 로그인한 모든 조종자에 대해 생성되며, 기본적으로 삭제되지 않는다.



[그림 3-8] DJI GO의 비행로그 동기화 로그 파일 내 조종자 정보 식별 결과

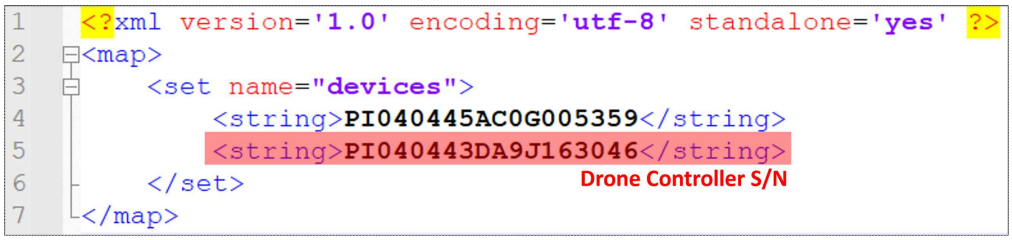
'/data/data/[package]/databases' 경로에는 데이터베이스 파일인 'flysafe_license_unlock.db' 파일이 존재한다. 해당 파일은 DJI Fly·Pilot·SC에서도 동일하게 존재하며, [그림 3-9]와 같이 'user_info' 테이블의 'token' 컬럼에 조종자의 인증 토큰을 저장하는 구조를 갖는다.



[그림 3-9] flysafe_license_unlock.db 파일 분석을 통한 조종자 정보 식별 결과

mmkv.default 파일은 DJI GO4·Fly와 DJI SC의 GO4·Fly에서 공통적으로 존재한다. DJI GO4와 DJI SC의 GO4에서는 Key가 'key_account_token'인 Value에 조종자의 인증 토큰이 Base64로 인코딩된 상태로 저장된다. DJI Fly의 mmkv.default 파일에는 Key가 'last_request_promotion_params'인 Value에 조종자의 인증 토큰이, 'deviceDataManagerSpDataKey'인 Value에는 조종기의 시리얼 번호가 기록된다. DJI SC의 Fly는 Key가 'key_last_fly_sn_for_data_ug'인 Value에 조종기의 시리얼 번호가 기록된다.

Parrot Freeflight 6의 데이터 중 조종자 관련 데이터는 '/data/data/com.parrot.freeflight6/shared_prefs' 경로 하위에 위치한 activation.xml, ff6Prefs.xml, account.xml, com.google.android.gms.signin.xml 파일에 저장된다. activation.xml 파일에는 조종기의 시리얼 번호가 [그림 3-10]와 같이 드론 시리얼 번호 이후에 기록된다.



[그림 3-10] Parrot Freeflight 6의 activation.xml 분석을 통한 조종기 시리얼 번호 식별 결과

ff6Prefs.xml 파일에서 Key가 'last_remote_control_uid'인 Value에는 마지

막으로 연결한 조종기의 시리얼 번호가 기록된다. 또한 account.xml 파일에서 Key가 'account'인 Value에는 조종자의 어플리케이션 닉네임이, 'date'인 Value에는 조종자가 어플리케이션에 마지막으로 로그인한 시각이 ms 단위의 유닉스 타임 스탬프 형태로 기록된다. 'com.google.android.gms.signin.xml' 파일에는 조종자의 Google OAuth 인증 토큰과 이메일 정보가 기록된다. Key가 'googleSigninAccount:[32자리 문자열]'인 Value의 tokenId를 Key로 가지는 Value가 OAuth 인증 토큰이며, Key가 'email'인 Value가 조종자의 이메일을 의미한다. 해당 파일에서 수집한 인증 토큰은 조종자가 Parrot Cloud를 통한 기본 로그인이 아닌, 구글 로그인을 사용한 경우 생성되며, 클라우드 포렌식 단계에서 이를 'id_token'으로 활용하여 Parrot의 인증 토큰을 발급받을 수 있다.

3) 비행로그 파일

드론 조종기에서 생성되는 비행로그 파일은 제조사별로 상이하며, 각 제조사의 조종기에 생성되는 비행로그를 통해 획득 가능한 정보는 [표 3-7]와 같다. 획득 가능한 정보는 비행 정보, 위치 정보, 기기 식별자로 분류되며, 제조사 별로 획득 가능한 정보에 일부 차이가 존재한다.

[표 3-7] 드론 제조사별 비행로그 파일에 저장된 주요 정보

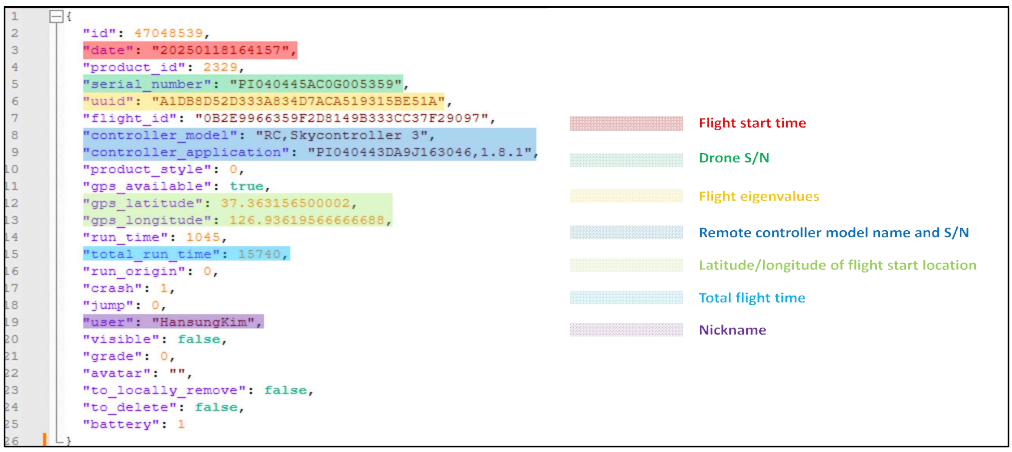
Manufacturer	Flight information	Location information	Device identifier
DJI	Flight speed, altitude, distance, time	- Drone flight path (latitude/longitude)	Drone S/N
Parrot	Flight speed, altitude, time	- Drone flight path (latitude/longitude) - Pilot location (latitude/longitude)	Drone S/N, RC S/N

DJI 드론의 비행로그 파일은 자체 포맷을 사용하는 TXT와 DAT 비행로그로 분류

된다. TXT 비행로그 파일은 조종기에서 생성되는 비행로그로, 'DJIFlightRecord_[yyyy-mm-dd]_[hh-mm-ss].txt' 형태의 파일 이름을 갖는다. DAT 비행로그 파일은 드론 기체에서 생성되는 비행로그 파일이며, 'yy-mm-dd-hh-mm-ss_FLY[Fly count].dat' 형태의 파일 이름을 가진다. 이는 드론 기체에서 DJI Assistant 2를 통해 추출할 수 있으며, DatCon과 같은 상용 도구를 통해 분석할 수 있다. DAT 비행로그는 모든 조종 어플리케이션 및 조종기에서 생성되는 TXT 비행로그와 다르게, DJI Fly, DJI SC에서만 생성된다. TXT 비행로그 파일에 대해 복호화 및 파싱 기능을 제공하는 다양한 상용 도구가 존재한다. 이러한 DJI 비행로그 파일로부터 드론의 비행 중 기록된 속도·고도·거리·시간 정보와 비행 경로, 비행을 수행한 드론의 시리얼 번호를 획득할 수 있다.

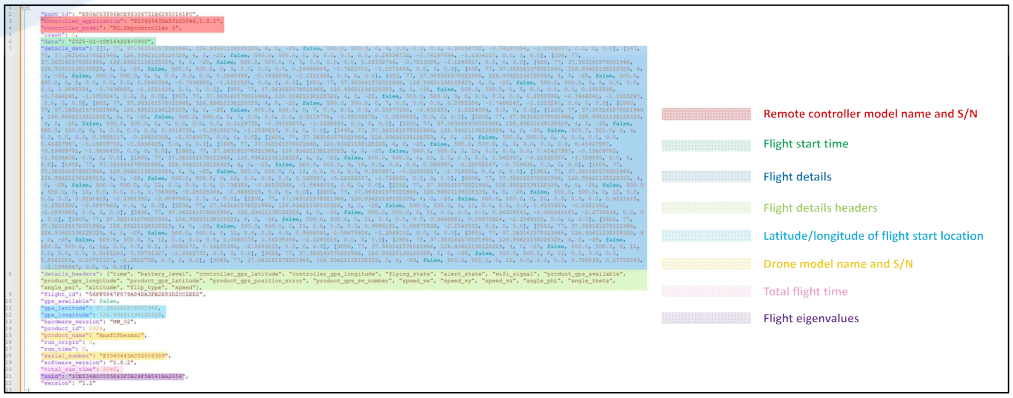
Parrot의 비행로그 파일은 JSON 유형의 파일인 비행로그 요약 파일과 비행로그 상세 파일로 구분된다. 비행로그 요약 파일은 해당 비행 기록에 대한 요약된 정보를 가지며, 비행로그 상세 파일은 드론의 이동 경로와 조종자의 위치 등 상세한 정보가 기록된 값이다. 두 유형의 파일명은 모두 '[len 4 number]_[yyyy]-[mm]-[dd]T[hhmmss]+0900_[len 6 string]_[fly count].json' 형태의 파일명을 갖는다. 그러나 비행로그 파일명이 동일하여도 생성되는 위치가 상이하다.

[그림 3-11]은 Parrot Freeflight 6의 비행로그 요약 파일의 분석 결과이다. 비행로그 요약 파일에서 획득 가능한 주요 정보는 드론의 비행 시작 시각과 위치, 드론과 조종기의 시리얼 번호이다. JSON 구조의 비행기록 파일에서 'date' 필드의 값에는 해당 비행이 시작된 날짜 및 시간에 대한 정보가 기록되며, 'total_run_time'은 해당 비행의 총 비행 시간이 ms 단위로 기록된다. 'serial_number'와 'controller_application'에는 비행을 수행한 드론과 조종기의 시리얼 번호가 기록되며, 'user'은 비행을 수행한 조종자의 어플리케이션 닉네임이다. 'gps_latitude'와 'gps_longitude'는 드론의 비행 시작 위치에 대한 위도·경도 값이며, uuid는 해당 비행에 대한 고유 ID를 의미한다. 이러한 비행로그 요약 파일은 '/data/data/com.parrot.freeflight6/files/ACADEMY/flightSummary' 경로에 저장된다.



[그림 3-11] Parrot Freeflight 6의 비행로그 요약 파일 분석 결과

[그림 3-12]은 Parrot Freeflight 6의 비행로그 상세 파일의 분석 결과이다. 비행로그 요약 파일과 동일하게 비행 시작 위치와 드론 및 조종기의 시리얼 번호와 같은 정보가 저장되며, details_data 필드에는 상세한 비행 기록이 존재한다. details_data 필드에는 21개의 데이터 유형이 정의되어 있는 2차원 배열이며, details_headers 필드에 각 인덱스가 의미하는 데이터가 정의되어 있다. 본 연구에서는 디지털 포렌식 관점에서 의의를 갖는 8개의 유형에 대한 분류를 진행한다.



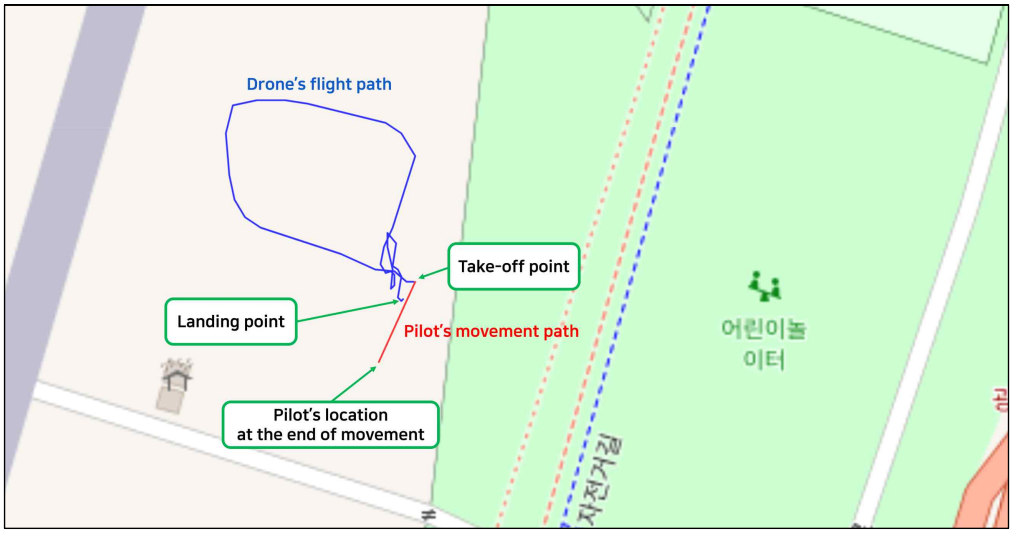
[그림 3-12] Parrot Freeflight 6의 비행로그 상세 파일 분석 결과

[표 3-8]은 details_headers 필드에 정의된 8가지 주요 데이터 유형에 대한 분석 결과이다. ‘time’은 비행 시각을 의미하는 ms 단위의 데이터로,

드론이 비행을 시작한 시점부터 경과된 시각을 의미한다. 이러한 데이터는 200ms 간격으로 기록되기 때문에, Parrot Freeflight 6의 비행로그 파일은 200ms 단위로 상세한 비행 로그가 기록된다. 'battery_level'은 현재 비행 시각에서의 드론의 배터리 잔량을 퍼센테이지 단위로 기록된 값이다. 'controller_gps_latitude'와 'controller_gps_longitude'는 조종자의 위치에 대한 위도·경도를 의미한다. 마찬가지로 'product_gps_longitude'와 'product_gps_latitude', 'altitude'는 드론의 비행 경로인 위도·경도와 고도를 의미한다. 이때 드론의 비행 경로는 드론의 GPS 사용 가능 여부에 따라 결정되며, 이에 대한 GPS 상태 정보는 'product_gps_available'에 기록된다. 이러한 details_data 내에 200ms 단위로 기록된 데이터는 드론의 비행을 통해 생성된 다양한 정보가 기록된다. 본 연구에서는 Parrot Freeflight 6의 비행로그 파일을 분석하여 [그림 3-13]와 같이 드론의 비행 경로와 비행 과정에서 조종자의 위치를 시각화하여, 실제 분석된 결과가 비행 실험을 통해 기록한 수치와 동일한지 여부를 검증하였다.

[표 3-8] Parrot Freeflight 6의 비행로그 상세 파일 분석 결과

details_headers	index of details_data	Data
time	0	Flight timestamp (in milliseconds)
battery_level	1	Battery level (in percentage)
controller_gps_ latitude	2	Remote controller GPS latitude
controller_gps_ longitude	3	Remote controller GPS longitude
product_gps_ available	7	Drone GPS availability
product_gps_ longitude	8	Drone GPS longitude
product_gps_ latitude	9	Drone GPS latitude
altitude	18	Drone altitude (in meters)



[그림 3-13] Parrot Freeflight 6의 비행로그 시각화 결과

DJI 비행로그 파일은 암호화가 적용되었으며, Parrot 비행로그 파일은 별도로 암호화가 적용되지 않는다. 본 연구에서는 DJI GO4의 TXT 비행로그 복호화에 대한 선행 연구와 DJI의 OpenAPI를 종합적으로 활용하여 TXT 비행로그에 대한 복호화를 수행하였다. 이에 대한 자세한 내용은 4절에서 설명한다.

4) 미디어 파일

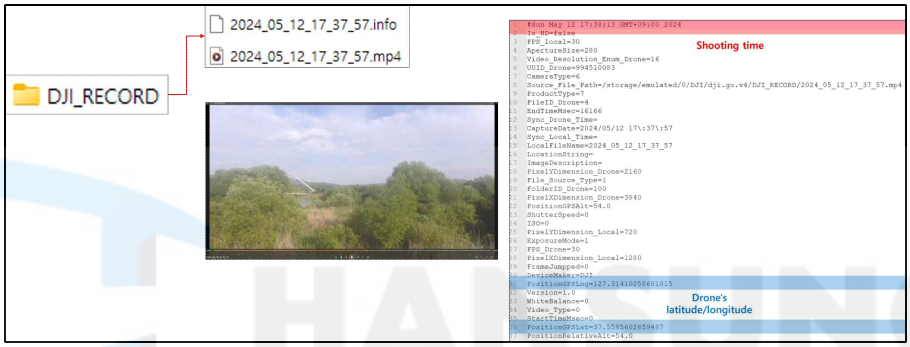
드론 조종기를 통해 드론의 촬영 기능을 사용할 경우, [표 3-9]와 같이 미디어 파일이 조종기에 생성된다. 이러한 미디어 파일은 이미지와 동영상으로 분류되며, 드론이 촬영한 데이터 외에도, 드론 조종기가 동작하면서 생성된 캐시 파일 또한 존재한다. Parrot Freeflight 6는 조종기를 통해 드론의 촬영 기능을 사용할 경우 조종기에 미디어 파일이 즉시 생성되지 않으며, 조종자가 별도로 드론 조종 어플리케이션을 통해 드론 기체에 존재하는 미디어 데이터를 다운로드 받아야 한다.

[표 3-9] 미디어 파일 분석 결과 분류

Application	File name	Acquirable Data
DJI GO	org_[16 character string]_ [13 digit number].jpg	Photos captured by the drone and capture dates
	org_[16 character string]_ [13 digit number].mov	Recorded videos and capture dates
	[yyyy]_[mm]_[dd]_[hh]_[mm]_[ss].mp4	Recorded video files
	[yyyy]_[mm]_[dd]_[hh]_[mm]_[ss].info	Capture dates of recorded videos
	thumb_[[16 character string]_ [13 digit number].jpg	Thumbnail images of captured photos and videos
	screen_[16 character string]_ [13 digit number].jpg	Cached photo files and capture dates
	screen_[16 character string]_ [13 digit number].info	Capture dates of captured photos
DJI GO4	thumb_[string]_[number].jpg	Thumbnail image files of captured photos and videos
	[yyyy]_[mm]_[dd]_[hh]_[mm]_[ss].info	Recording date, video duration, location information, and video resolution of recorded videos
	[yyyy]_[mm]_[dd]_[hh]_[mm]_[ss].mp4	Recorded video files
DJI Fly	[yyyy_mm_dd_hh_mm_ss]_Cache.mp4	Cached recorded video files
	FD_h264cache/[Drone serial number]/thumbnail.jpeg	Image files captured during drone boot
	Photo_[number]_ [media file name]_[number]_0_ [Captured timestamp]_ photo_thumbnail.jpg	Thumbnail files of captured photos
	Video_[number]_ [media file name]_[number]_0_ [Captured timestamp]_ photo_thumbnail.jpg	Thumbnail files of recorded videos
	dji_fly_YYYYMMDD_HHMMMSS_Number_UnixTime(millisecond)_photo.jpg	Photos captured by the drone

	dji_fly_YYYYMMDD_HHMMMSS_Number_UnixTime(millisecond)_video.mp4	Video files recorded by the drone
DJI Pilot	org_[16 character string]_ [13 digit number].jpg	Photos captured by the drone
	org_[16 character string]_ [13 digit number].mov	Videos recorded by the drone
	thumb_[16 character string]_ [13 digit number].jpg	Thumbnails of photos and videos captured by the drone
DJI SC (DJI GO4)	[yyyy_mm_dd_hh_mm_ss].mp4	Video files recorded by the drone
	[yyyy_mm_dd_hh_mm_ss].info	Recording date/time and location information for video files
	CACHE_IMAGE/pov_*	Cached image files
	thumb_[16 character string]_ [Unix timestamp].jpg	Cached image files
DJI SC (DJI Fly)	Photo_[number]_ [media file name]_[number]_0_ [Captured timestamp]_ photo_thumbnail.jpg	Thumbnail files of captured photos
	Video_[number]_ [media file name]_[number]_0_ [Captured timestamp]_ photo_thumbnail.jpg	Thumbnail files of recorded videos
	OriginalFiles/Photo/*.jpg	Captured photo files
	DJI FLY/Video/*.mp4	Recorded video files
DJI SC (DJI Pilot)	CACHE_IMAGE/*.jpg	Cached thumbnail image files
	org_[16 character string]_ [Unix time stamp].jpg	Photo files captured by the drone
	org_[16 character string]_ [Unix time stamp].mp4	Video files recorded by the drone
Parrot Freeflight 6	[Drone model name]/ [Drone serial number] /single-photo/*JPG	Photo files captured by the drone
	[Drone model name]/ [Drone serial number]/video/*MP4	Video files recorded by the drone

DJI 드론 조종 어플리케이션의 미디어 파일 중 info 확장자를 가진 파일은 동일한 파일명을 가진 JPG 또는 MP4 파일에 대해 촬영 시각과 위치와 같은 정보를 저장하는 텍스트 파일이다. DJI GO4 어플리케이션에서의 예시로 [그림 3-14]와 같이 로컬 환경에 동일한 이름을 가진 info 파일과 MP4 파일이 존재한다. 이때 MP4 파일은 드론이 촬영한 동영상 파일이며, info 파일에서는 드론이 해당 동영상 파일을 촬영한 시각과 위치 좌표가 기록된다.



[그림 3-14] DJI GO4에서의 동영상 파일과 info 파일 분석 결과

Parrot Freeflight 6의 미디어 파일에는 Exif 메타데이터에 다양한 정보가 포함된다. [표 3-10]는 Parrot Freeflight 6를 통해 드론을 조종하여 촬영한 사진 및 동영상 파일에 대한 Exif 메타데이터 분석 결과이다. JPG 파일과 MP4 파일에 대해 메타데이터를 통해 획득 가능한 주요 정보는 촬영 시각, 드론의 모델명과 시리얼 번호이다.

[표 3-10] Parrot Freeflight 6의 미디어 파일 메타데이터 분석 결과

File Type	Name	Data	Value example
JPG	Create Date	File Creation Time	2025:02:23 13:23:05
	Image Description	Drone model name and software version	AnafiThermal, 1.8.2
	Vehicle uuid	Drone S/N	PI040445AC0G005359
	Serial number	Drone S/N	PI040445AC0G005359

MP4	Content Create Date	File Creation Time	2025:02:23 13:23:14+09:00
	Make	Drone manufacturer name	Parrot
	Model	Drone model name	AnafiThermal
	Software	Drone software version	1.8.2
	Parrot Serial	Drone S/N	PI040445AC0G005359

5) 어플리케이션 로그 파일

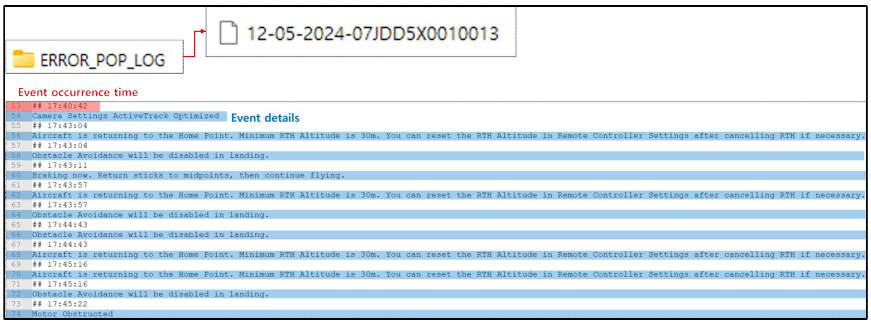
드론 조종 어플리케이션이 동작하는 과정에서 로그 파일을 생성하며, 이러한 로그 파일에는 비행 중 발생한 이벤트와 드론·조종기 시리얼 번호와 같은 정보가 저장된다. [표 3-11]는 DJI 드론 조종 어플리케이션에서 식별할 수 있는 로그 파일의 분석 결과이다. 파일명은 로그 파일의 생성 시각이 포함되며, 일부 로그 파일은 로그 파일을 생성한 드론의 시리얼 번호가 포함된다.

[표 3-11] 드론 조종 어플리케이션 로그 파일 분석 결과

Application	File name	Data
DJI GO4	ERROR_POP_LOG/[dd]-[mm]-[yyyy]-[Drone S/N]	In-flight drone event logs
DJI Fly	AccountCenter/log-[yyyy]-[mm]-[dd].log	Cached user authentication tokens
	deviceManager/log-[yyyy]-[mm]-[dd].log	Serial numbers and binding information of the drone and RC controller
	AeroScopeLogic/log-[yyyy]-[mm]-[dd].log	Latitude and longitude of the DJI RC
DJI SC(DJI GO4)	ERROR_POP_LOG/[dd]-[mm]-[yyyy]-[Drone S/N]	In-flight drone events
DJI SC(DJI Fly)	deviceManager/log-[yyyy]-[mm]-[dd].log	Serial numbers and binding information of the drone and RC controller
	AeroScopeLogic/log-[yyyy]-[mm]-[dd].log	Latitude and longitude of the drone controller

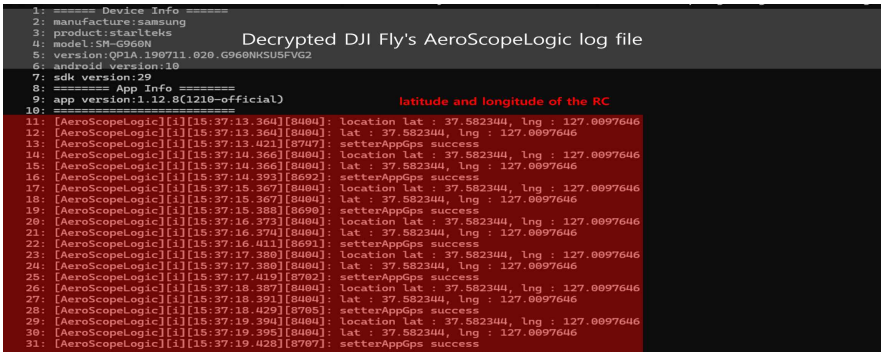
DJI GO4 어플리케이션은 '/sdcard/DJI/dji.go.v4/LOG/ERROR_POP_

LOG’ 경로 하위에 ‘[dd]-[mm]-[yyyy]-[Drone S/N]’ 형태로 로그 파일을 저장한다. 해당 로그 파일에는 특정 시리얼 번호를 가진 드론이 비행 중 발생한 이벤트가 평문 텍스트 형태로 기록된다.



[그림 3-15] DJI GO4의 ERROR_POP_LOG 분석 결과

DJI Fly 어플리케이션은 ‘/sdcard/Android/data/dji.go.v5’ 경로 하위에 위치한 AccountCenter, deviceManager, AeroScopeLogic 디렉터리에 주요한 로그 파일을 저장한다. 로그 파일은 ‘log-[yyyy]-[mm]-[dd].log’ 형식의 파일명을 가지며, 내부 데이터는 암호화 및 Base64 인코딩이 적용된 형태로 저장된다. AccountCenter 로그 파일은 조종자의 인증 토큰이 저장되며, deviceManager 로그 파일은 드론 및 조종기의 시리얼 번호와 계정에 바인딩된 장치 정보가 기록된다. AeroScopeLogic 로그 파일은 [그림 3-16]와 같이 드론의 비행 과정에서 기록된 조종기의 위도·경도가 저장된다.



[그림 3-16] DJI Fly의 AeroScopeLogic 로그 파일 분석 결과

DJI SC의 로그 파일은 DJI SC에서 동작하는 DJI GO4·Fly 어플리케이션의 앱 데이터에 존재한다. DJI SC의 DJI GO4 어플리케이션은 모바일 환경의 DJI GO4 어플리케이션과 로그 파일 구조 및 내용이 동일하며, DJI Fly 어플리케이션은 AccountCenter 로그 파일을 제외한 deviceManager 및 AeroScopeLogic 로그 파일이 동일하게 존재한다.

모바일 및 DJI SC에서 동작하는 DJI GO4·Fly 어플리케이션은 앞서 서술한 바와 같이 암호화와 Base64 인코딩이 적용되기에, 로그 파일 내용을 분석하기 위해선 복호화 과정이 선행되어야 한다. 본 연구에서는 모바일 환경에서의 DJI GO4·Fly 어플리케이션에 대한 역공학 분석을 통해 [수식 3-4]와 같이 복호화 알고리즘을 식별하였다. 암호 키는 고정된 문자열인 'DJILog@SimpleEncryption'을 입력 값으로 하는 PBKDF2-HMAC-SHA1 알고리즘을 통해 생성하며, IV는 고정된 값을 사용한다.

$$P = AES128 - CBC - PKCS5Padding_{decryption}(\mathrm{Base64}_{\mathrm{decode}}(\mathrm{C}))$$

$$K = PBKDF2 - HMAC - SHA1(P = DJILog@SimpleEncryption,\\ S = 0xc8570ac98cc615aa6a6b97b3f20f1b41, c = 1{,}000, dkLen = 16)$$
$$IV = 0x9d6c5cab5b0281255a222d1c861ddfdf$$

[수식 3-4] DJI 앱 로그 파일 복호화 알고리즘

제 3 절 클라우드 데이터 분석

드론 조종기에서 동작하는 드론 조종 어플리케이션은 클라우드 서버와의 통신을 통해 데이터를 동기화한다. 이러한 구조로 인해 앞서 [표 3-2]와 [표 3-3]에서 식별한 바와 같이, 제조사의 클라우드 API를 통해 다양한 데이터를 수집할 수 있다. 이러한 정보는 조종자 정보, 드론 정보, 비행로그 파일로 분류된다.

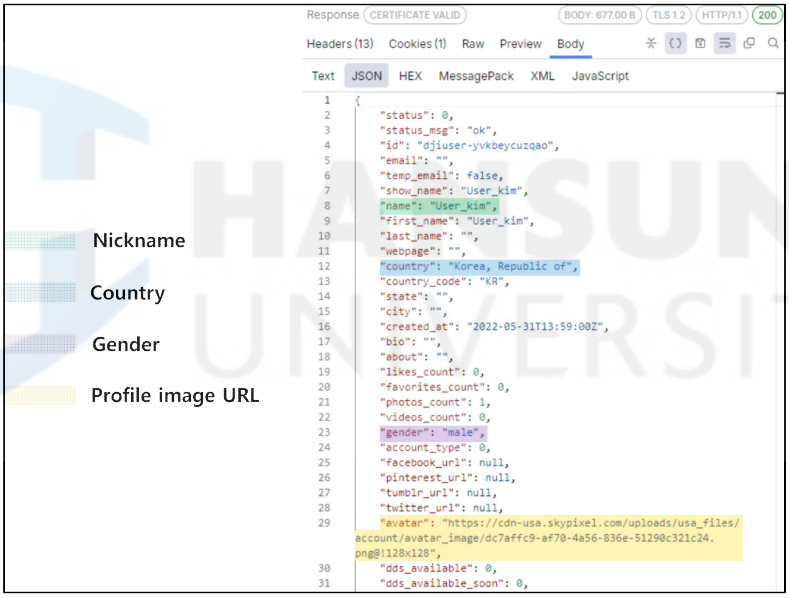
[표 3-12]는 DJI 클라우드의 주요 API에 대한 응답 데이터를 분석한 결과이다. 응답 데이터는 PNG, JSON, Zlib 압축 파일로 분류된다.

[표 3-12] DJI 클라우드 API의 응답 데이터 분석 결과

API number	Purpose	Response data	Response type
API-1	Get Captcha Image	Captcha image	PNG
API-2	Get Token	Logged-in pilot information	JSON
API-3	Get Pilot Information	Application profile information of the logged-in pilot	JSON
API-4	Get Connected Drone Information	Information on the drone that the pilot connected to the remote control and performed the flight (S/N, total number of flights/time/distance)	JSON
API-5	Get Flight Record List	Flight record files generated by DJI GO/GO4/Pilot	JSON
API-6	Get Flight Record List	Flight record files generated by DJI Fly	JSON
API-7	Get Flight Record Download	TXT flight record file compressed with zlib	Zlib file
API-8	Convert S/N	By querying the S/N of the device received as a request, the results of identifying various S/Ns of the device are displayed	JSON
API-9	Get Flight History Information	Flight history of a specific drone	JSON
API-10	Get Pilot Binding Information	List of drones bound to the pilot's account	JSON
API-11	Get Drone Binding Information	List of controllers bound to the drone	JSON

API-1과 API-2는 로그인 과정에서 사용되는 API이다. API-1은 캡차 인증을 위해 PNG 형식의 캡차 이미지를 다운로드 하는 API로, 응답 데이터로 4자리 인증 문구가 기록된 이미지 파일을 획득할 수 있다.

API-2는 캡차 인증을 통해 로그인을 수행하는 API로, 응답 데이터로 로그인한 조종자의 인증 토큰을 획득할 수 있다. 이러한 응답 데이터는 조종자의 이메일, 고유 ID, 어플리케이션 닉네임 등의 정보가 포함된다. API-3는 로그인 후, 조종자의 어플리케이션 프로필을 조회할 때 사용되는 API이다. 획득 가능한 정보로 [그림 3-17]와 같이 조종자의 닉네임, 국가, 성별, 프로필 사진 URL 등이 존재한다. API-4의 응답 데이터는 조종자가 드론 조종 어플리케이션에 연결하여 비행을 수행한 드론에 대한 정보이다. 이러한 데이터에는 드론의 시리얼 번호뿐만 아니라, 해당 드론이 비행을 수행한 총 비행 횟수, 시각, 거리 정보가 포함된다.



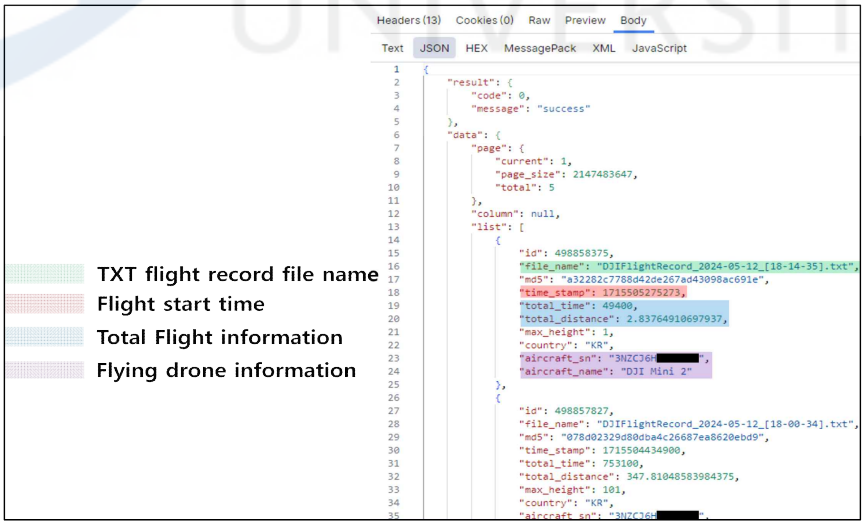
[그림 3-17] DJI 클라우드의 API-3 응답 데이터 분석 결과

API-5와 API-6의 응답 데이터는 드론 조종 어플리케이션에서 생성된 TXT 비행로그 파일 목록이다. 이때 API-5는 DJI GO·GO4·Pilot에서 생성된 비행로그 파일이며, API-6는 DJI Fly에서 생성된 비행로그 파일이다. API-5의 응답 데이터는 [그림 3-18]와 같이 비행로그 파일명을 목록화한 형태이며, API-6는 [그림 3-19]와 같이 비행로그 파일명과 비행 시작 시각, 해당 비행을 수행한 드론의 정보, 비행에 대한 요약된 정보

를 포함한다. API-7은 이러한 API-5, 6를 통해 조회한 특정 비행로그 파일을 다운로드 하는 기능을 수행한다. API-7에 대한 요청 패킷에 비행로그 파일명을 포함하며, 응답 데이터는 해당 비행로그 파일이 zlib 형식으로 압축된 데이터를 반환한다.



[그림 3-18] DJI 클라우드의 API-5 응답 데이터 분석 결과



[그림 3-19] DJI 클라우드의 API-6 응답 데이터 분석 결과

API-8은 특정 시리얼 번호를 가진 장치의 다른 시리얼 번호를 조회하는

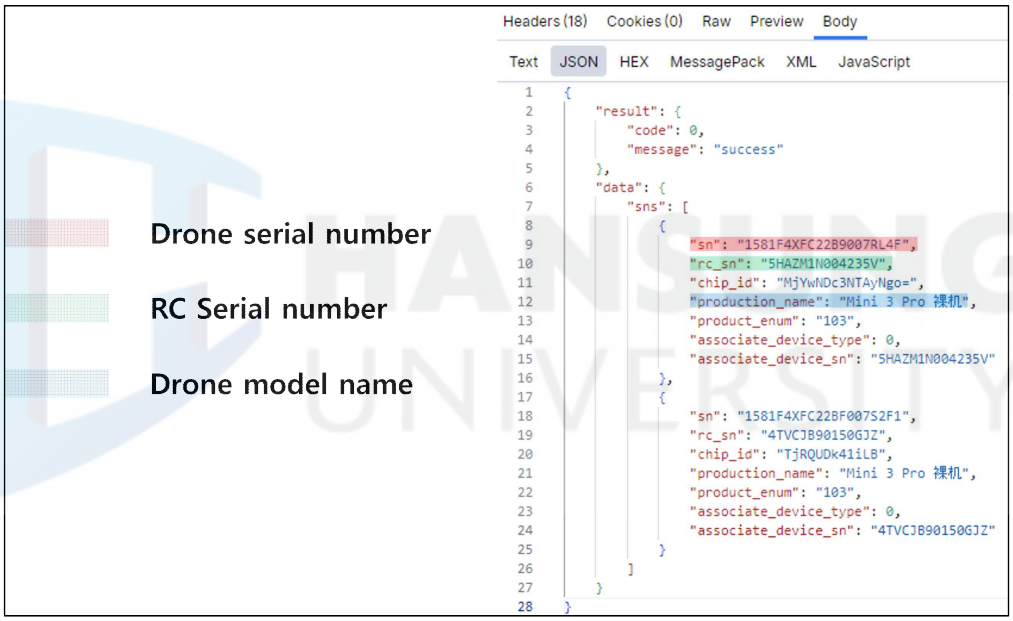
기능을 수행한다. DJI의 드론을 가리키는 시리얼 번호는 배터리, 카메라, 비행컨트롤러, 기체로 분류된다. [그림 3-20]는 드론 조종 어플리케이션에 드론을 연결 후, 드론 조종을 위한 카메라 뷰에서 연결된 드론의 시리얼 번호를 확인한 결과이다. 이러한 시리얼 번호 중 드론 기체를 나타내는 시리얼 번호는 '기체 SN'과 '비행 컨트롤러 SN'이다. 본 연구에서 DJI 드론을 대상으로 로컬 아티팩트 및 클라우드 데이터를 분석한 결과, 특정 드론을 나타내는 시리얼 번호가 혼재되어 사용된다. API-8은 이러한 드론의 특정 S/N를 전송 시, 해당 시리얼 번호를 가진 드론의 '카메라 SN', '기체 SN', '비행 컨트롤러 SN'를 반환한다. 본 연구에서는 데이터 분석 과정에서 시리얼 번호의 혼재를 막기 위해 분석 과정에서 식별한 드론을 가리키는 시리얼 번호를 API-8을 통해 '기체 SN'로 통일하였다.



[그림 3-20] DJI 드론 조종 어플리케이션을 통한 드론 시리얼 번호 식별 결과

API-9는 드론의 시리얼 번호를 통해, 해당 드론의 비행 이력을 조회하는 기능을 수행한다. 해당 API는 조회한 드론의 총 비행 거리 및 시간 정보를 반환하며, 이를 통해 특정 드론이 실제 비행 이력이 있는 드론인지 여부를 식별할 수 있다. API-10과 API-11은 바인딩 정보를 조회하는 기능을 수행한다. DJI 드론에서 바인딩이란 조종자의 계정에 드론을 연결하거나, 드론에 조종기를 연결하는 기능을 수행한다. 이러한 바인딩은 DJI

서비스를 이용하는 조종자에게 드론 또는 조종기와의 소유 및 연동 정보를 기록함으로써, 비행 중 기체가 분실되는 사고 발생 시 DJI Care 플라이어웨이 보장을 청구할 수 있는 공식 기능이다. 이러한 바인딩은 단순히 기기를 연결하여 사용하는 페어링과는 구분된다. [그림 3-21]은 API-10을 통해 조종자의 계정에 바인딩된 기기 정보를 조회한 결과이다. 기본적으로 조종자의 계정에 바인딩된 드론의 시리얼 번호 및 모델명이 sn와 production_name 필드에 기록되며, 드론에 바인딩된 조종기가 있는 경우 조종기의 시리얼 번호가 rc_sn 필드에 존재한다.



[그림 3-21] 조종자의 계정에 바인딩된 기기 정보 조회

[그림 3-22]는 API-11을 통해 드론에 바인딩된 계정 및 조종기 정보를 조회한 결과이다. 드론에 바인딩된 조종기는 rc_sn 필드에 시리얼 번호가 존재하며, 바인딩된 조종기가 없을 경우 해당 필드가 빈 값으로 표시된다. 조종자의 계정 정보는 bind_account 필드에 존재하며, 일부가 마스킹된 상태로 존재한다.



[그림 3-22] 드론에 바인딩된 계정 및 조종기 정보 조회

[표 3-13]은 Parrot 클라우드 API에 대한 응답 데이터를 분석한 결과이다. Parrot 클라우드는 DJI와는 다르게 응답 데이터의 유형이 모두 JSON 형태이다.

[표 3-13] Parrot 클라우드 API의 응답 데이터 분석 결과

API number	Purpose	Response data	Response type
API-12	Get Token	Pilot's authentication token	JSON
API-13	Get Token	Pilot's authentication token	JSON
API-14	Get Pilot Information	Application profile information of the logged-in pilot	JSON
API-15	Get Flight record file	Flight record summary file	JSON
API-16	Get Flight record file	Flight record detail file	JSON

API-12와 API-13은 Parrot 클라우드 로그인 기능을 위한 API로, 각각 Parrot 크리덴셜과 Google OAuth 크리덴셜을 요청 패킷에 구성 시 Parrot 클라우드 서버와의 통신을 위한 인증 토큰을 반환한다. 응답 데이터는 두 가지 API 모두 동일한 구조를 갖는 JSON 형식으로, 인증 토큰은 응답 JSON 데이터의 apcToken 필드에 존재한다. 해당 인증 토큰은

API-14, 15, 16에 대한 클라우드 API 통신 시 조종자 인증을 위해 사용된다. API-14는 조종자의 Parrot Freeflight 6 어플리케이션 프로필 정보를 조회하는 기능을 수행한다. 응답 데이터로 획득할 수 있는 주요 정보는 조종자의 닉네임과 이메일이다. API-15와 API-16은 [그림 3-23]와 같이 JSON 형태의 비행로그 파일을 수집하는 API이다. API-15은 조종자가 생성하여 클라우드에 동기화된 비행로그 요약 파일을 목록화된 형태로 제공한다. 따라서 클라우드에 동기화된 조종자의 모든 비행로그 파일은 해당 API를 통해 식별할 수 있다. API-15를 통해 수집한 비행로그 요약 파일은 로컬 아티팩트에 존재하는 비행로그 요약 파일과 일부 상이한 구조를 갖지만 디지털 포렌식 관점에서 의미있는 데이터는 동일하게 갖는다. API-16은 비행로그 상세 파일을 제공하며, 해당 API에 대한 요청 패킷에 API-15에서 조회된 특정 비행로그 요약 파일의 id 필드 값을 요구한다. API-16을 통해 조회된 JSON 형태의 비행로그 상세 파일은 로컬 아티팩트에 존재하는 비행기록 상세 파일과 완전히 동일한 구조를 갖는다.

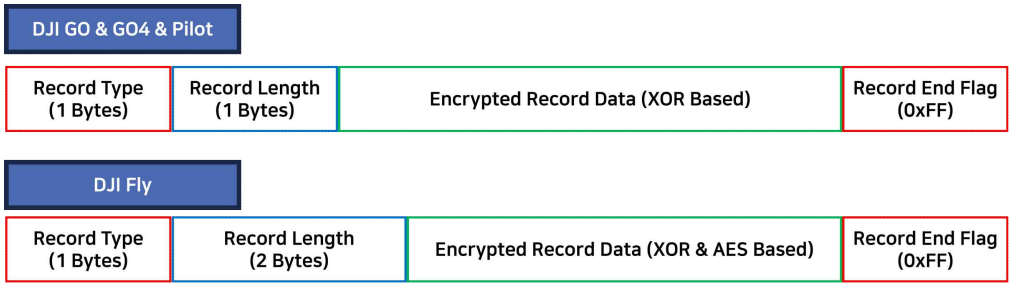


[그림 3-23] Parrot 클라우드 API를 통해 수집한 비행로그 파일 분석 결과

제 4 절 비행로그 파일 복호화

DJI 드론 조종 어플리케이션을 통해 생성되는 TXT 비행로그 파일은 DJI의 자체적인 포맷을 통해 저장된다. 이러한 드론 조종 어플리케이션의 비행로그 파일 포맷은 어플리케이션 유형별로 상이하지만, 모바일 및 DJI SC로 구분되는 동작 환경에서는 차이가 없다. 본 연구에서는 DJI GO4의 비행로그 파일에 대한 선행 분석 연구 결과를 활용하여 DJI GO·GO4·Pilot의 비행로그 파일이 동일한 암호화 방식을 사용하는 것을 식별하였으며, DJI의 OpenAPI를 기반으로 DJI Fly에서 생성된 TXT 비행로그 파일 복호화하였다.

[그림 3-24]는 DJI 어플리케이션 4종의 TXT 비행로그 파일 구조이다. TXT 비행로그 파일은 파일의 헤더와 복수의 레코드로 구성되어 있으며, 각 레코드는 암호화된 상태로 존재한다. 따라서 암호화된 레코드를 복호화하여 데이터를 획득할 수 있으며, 각 레코드에 저장되는 데이터는 레코드의 유형별로 상이하다. 비행로그 파일의 레코드는 최상위에 레코드의 유형을 나타내는 1바이트의 Record Type과 레코드의 끝을 의미하는 1바이트의 Record End Flag가 위치한다. Record Type 이후에는 레코드의 길이를 의미하는 Record Length와 암호화된 레코드 데이터인 Encrypted Record Data가 존재한다. DJI GO·GO4·Pilot 어플리케이션의 TXT 비행로그 파일은 1바이트의 Record Length를 가지며, DJI Fly 어플리케이션은 2바이트의 Record Length를 갖는다. 두 유형의 비행로그 파일은 Encrypted Record Data의 암호화 방식에서도 차이가 존재한다. DJI GO·GO4·Pilot에서는 XOR 기반의 암호화만 적용되며, DJI Fly에서는 XOR 기반 암호화와 AES 암호화가 함께 적용된다.

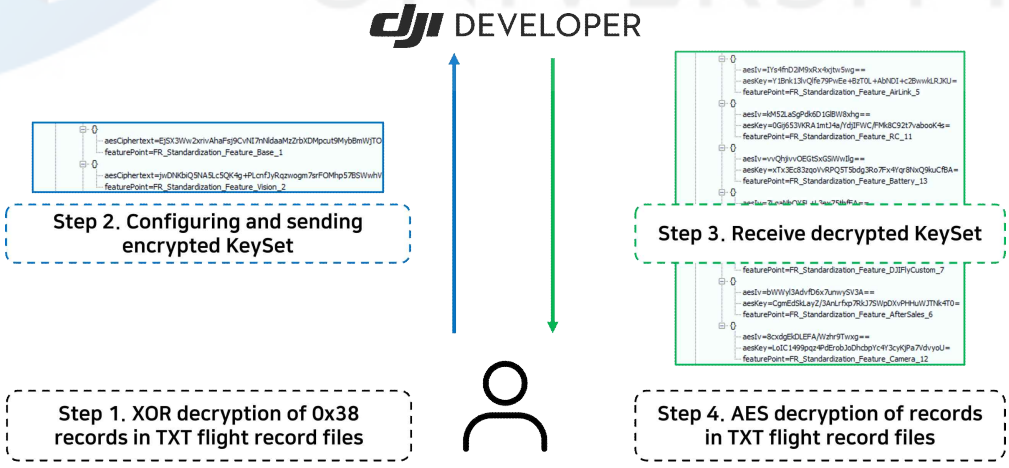


[그림 3-24] DJI 드론 앱의 TXT 비행로그 파일 구조

XOR 기반 암호화는 8바이트 길이의 키 스트림이 주기적으로 반복되어 각 바이트에 XOR 연산이 수행되는 형태로 진행된다. 해당 방식은 동일한 키 스트림을 동일 위치에 다시 XOR 연산을 수행할 경우 원문이 복원되는 대칭 구조를 갖는다. 복호화 과정의 핵심은 각 레코드에 대응하는 8바이트 scramble bytes 키 스트림을 정확히 재연하는 것이다. 선행 연구에서 제시된 방식에 따르면, 키 스트림은 레코드 타입과 암호화된 레코드 데이터의 첫 바이트(Key byte)를 입력으로 하여 CRC64 연산을 수행함으로써 생성된다. 즉, 레코드 타입과 첫 바이트를 결합한 2바이트 입력에 대해 사전에 정의된 CRC64 다항식과 초기 값을 사용하여 8바이트 CRC 값을 계산하여 scramble bytes로 사용한다. 이러한 구조로 인해 키 스트림은 레코드의 구조적 속성과 Encrypted Record Data의 첫 바이트에 종속적으로 결정되며, 레코드가 동일한 데이터를 가진 경우에도 초기 바이트가 달라진 경우 서로 다른 키 스트림이 생성된다. 생성된 8바이트 키 스트림은 Encrypted Record Data에 대해 XOR 연산이 순환 적용된다.

DJI Fly에서는 XOR 기반 암호화뿐만 아니라 AES 기반의 암호화를 추가적으로 활용한다. [그림 3-25]은 DJI Fly의 TXT 비행로그 파일을 복호화 하기 위한 절차이다. DJI Fly의 Encrypted Record Data를 복호화하기 위해선 AES Key와 IV가 요구되며, 이는 DJI Fly의 TXT 비행로그에서 0x38로 시작하는 KeySet 레코드에 암호화된 상태로 존재한다. 암호화된 KeySet은 0x38을 Record Type으로 가지며, 2바이트의 레코드 길이, 마지막 바이트에 0xFF가 위치한 구조이다. 본 연구에서는 앞선 XOR 복호화 과정을 KeySet 레코드를 대상으로 적용하여 복호화를 수행하였다.

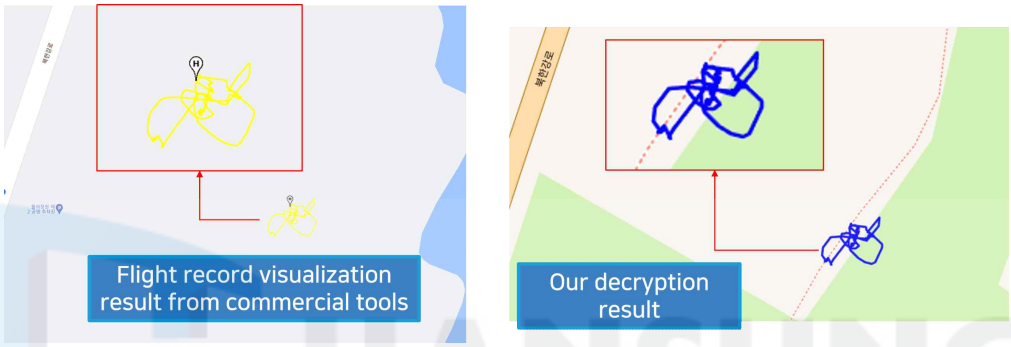
XOR 복호화가 완료된 KeySet 레코드에서 0번째 바이트는 featurePoint 값으로 해석되며, 해당 값은 DJI 개발자 API에서 요구하는 문자열 식별자로 매핑된다. 또한 XOR 복호화 결과 중 4번째 바이트 이후에 오는 값은 AES 키체인 암호문으로 간주되며, 해당 바이트열을 Base64로 인코딩하여 aesCiphertext 필드로 구성한다. 최종적으로 KeySet 레코드로부터 추출된 '{featurePoint, aesCiphertext}' 쌍을 JSON 형태로 구성하여 DJI 개발자 API인 'https://dev.dji.com/openapi/v1/flight-records/keychains'로 HTTP POST 요청을 전송한다. 해당 API 요청의 응답으로 각 featurePoint에 대응하는 AES 키와 IV가 Base64 인코딩이 적용된 문자열 형태로 반환되며, 이를 디코딩함으로써 레코드 복호화에 사용 가능한 암호 키 및 IV를 확보할 수 있다. 이후 일반 레코드의 복호화는 두 단계로 구성된다. 첫 번째 단계로 각 레코드의 암호화 데이터에 대해 KeySet 레코드와 동일하게 XOR 복호화를 수행한다. 두 번째 단계는 XOR 복호화가 적용된 값에 대해, RecordType에 대응하는 featurePoint를 기준으로 선택된 암호키 및 IV를 사용하여 AES256-CBC-PKCS5Padding 복호화를 수행함으로써 최종적으로 평문 레코드를 획득한다.



[그림 3-25] OpenAPI를 활용한 DJI Fly의 비행로그 복호화 절차

본 연구에서는 앞서 서술한 방식을 통해 DJI 어플리케이션에서 생성된

TXT 비행로그 파일을 복호화하였으며, 복호화 결과를 검증하기 위해 상용 도구의 분석 결과와의 비교를 진행하였다. TXT 비행로그 파일의 복호화된 레코드 중 Record Type이 0x01인 레코드에 저장된 드론의 비행 경로를 시각화하였다. 이후 상용 도구에서 시각화한 드론의 비행 경로와 시각적으로 비교하였으며, [그림 3-26]와 같이 동일한 결과가 나타나는 것을 확인하였다.



[그림 3-26] DJI 드론 앱의 TXT 비행로그 파일에 대한 복호화 시각화 결과 비교

HANSUNG UNIVERSITY

제 4 장 드론 포렌식 통합 분석

제 1 절 드론-조종자 상관성 분석

드론과 조종기 사이의 상관성을 증명하는 것은 드론 포렌식의 핵심 기술이다. 본 연구에서는 드론 기체와 조종기, 조종자의 계정을 수사 과정에서 획득한 상황에서 각 요소의 상관성을 증명한다. 이 과정에서 조종기에 저장된 로컬 아티팩트에서 드론의 시리얼 번호를 식별하고, 실제 드론 기체에서 식별할 수 있는 시리얼 번호를 비교한다.

1) 드론 시리얼 번호 식별

DJI와 Parrot의 드론 기체에는 시리얼 번호가 표기되어 있으며, 이는 해당 장치에 대한 고유 식별자 역할을 한다. 디지털 포렌식 관점에서 시리얼 번호는 범죄와 관련된 드론을 특정하기 위해 활용할 수 있으며, 조종기에 기록된 시리얼 번호를 통해 드론과 조종기간 상관성을 증명하기 위한 단서를 제공한다. 본 연구에서는 [그림 4-1] 및 [그림 4-2]와 같이 실제 연구에 사용된 드론 기체에서 시리얼 번호를 식별하였다.



[그림 4-1] DJI 드론 기체에서의 시리얼 번호 식별 결과



[그림 4-2] Parrot 드론 기체에서의 시리얼 번호 식별 결과

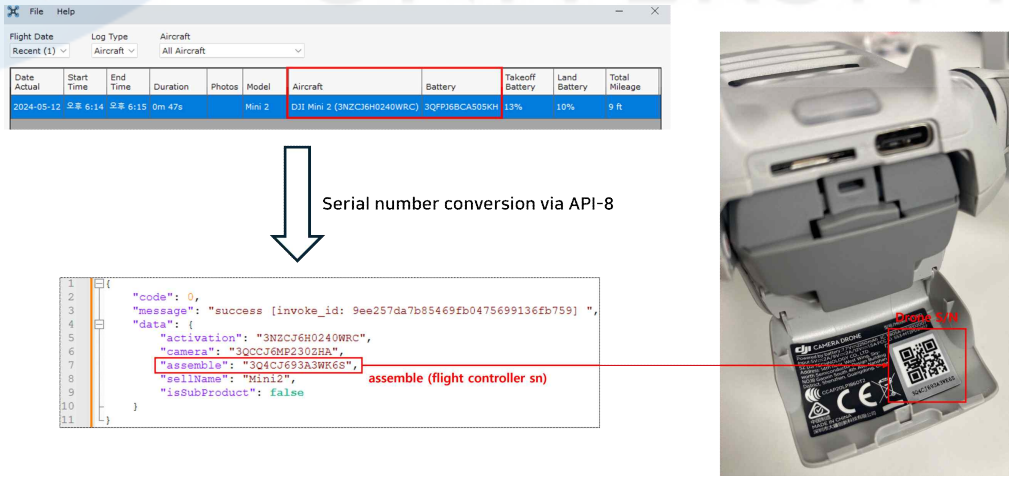
DJI 드론의 시리얼 번호는 드론의 구성 장치별로 부여되어 있다. 앞서 3장 2절에서 서술한 바와 같이, 드론 기체를 나타내는 시리얼 번호는 '기체 SN'과 '비행 컨트롤러 SN'로 분류된다. 드론 조종 어플리케이션과 클라우드에는 이러한 드론 시리얼 번호가 혼재되어 사용되기에, 본 연구에서는 API-8를 통해 모든 DJI 드론의 시리얼 번호를 '기체 SN'로 변환하여 활용한다.

Parrot 드론의 시리얼 번호는 드론 기체에 표기된 시리얼 번호와 로컬 및 클라우드 데이터에 저장되는 시리얼 번호의 형식이 상이하다. Parrot 드론 기체에는 'PS'로 시작하는 18자리 문자열로 시리얼 번호가 구성되어 있으며, Parrot Freeflight 6를 통해 생성된 데이터에 저장된 드론의 시리얼 번호는 'PI'로 시작하는 18자리 문자열로 시리얼 번호가 구성되어 있다. PS·PI 접두어가 포함된 문자열을 통해 각 시리얼 번호의 출처를 구분할 수 있으나, 각 출처에서 식별되는 시리얼 번호는 PS·PI 접두어 뿐만 아니라 전체 문자열 구성에서도 차이를 보인다. 드론 기체의 시리얼 번호

와 조종기의 로컬 아티팩트에 저장된 드론 시리얼 번호가 동일한 드론을 나타내는 경우, 하위 10자의 문자열이 동일하다. 이는 Parrot 드론 환경에서 갖는 시리얼 번호의 특성이다.

2) 상관성 분석

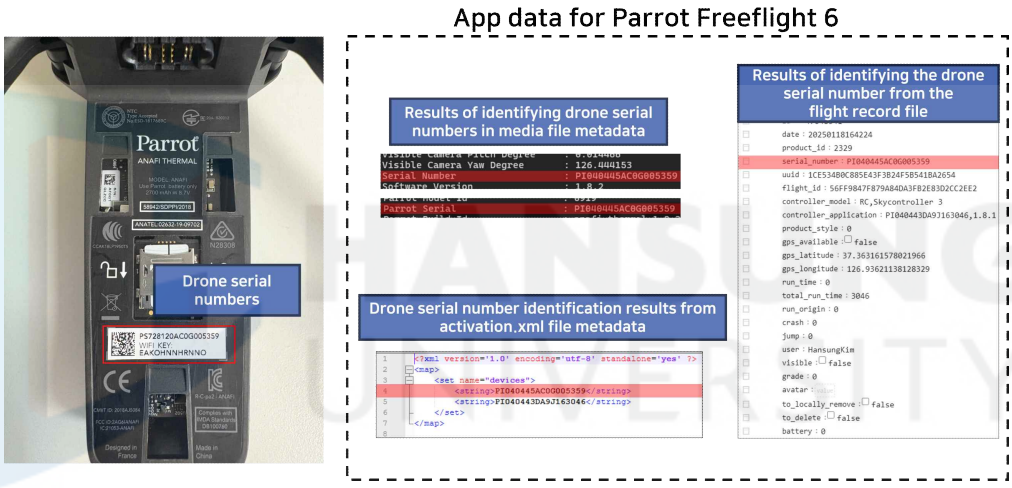
상관성 분석은 드론과 조종자의 상관성을 증명하는 과정이며, 이러한 과정은 드론의 시리얼 번호를 중심으로 진행된다. DJI 드론의 클라우드 서버에 저장된 데이터와 조종기의 드론 조종 어플리케이션 데이터에는 조종자가 연결하여 비행을 수행한 드론의 시리얼 번호가 기록된다. 본 연구에서는 [그림 4-3]와 같이 조종자 측면의 데이터에서 획득한 드론 시리얼 번호를 API-8을 통해 통일된 시리얼 번호 형식으로 변환한다. 이후 실제 드론 기체에 표기된 시리얼 번호와 비교하여 동일한 값을 가지는지 여부를 확인한다. 이러한 과정을 통해 특정 드론 기체에 대해 상관성 분석 대상 조종자의 계정 또는 조종기를 통해 비행을 수행하였는지 여부를 확인할 수 있다.



[그림 4-3] DJI 드론 환경에서의 상관성 분석 과정

Parrot 드론의 상관성 분석 또한 앞선 DJI 드론과 동일하게 시리얼 번

호를 비교하는 방식으로 진행한다. 그러나 시리얼 번호의 비교 방식에서 차이가 존재한다. [그림 4-4]는 Parrot 드론 기체와 드론 조종 어플리케이션 데이터에서의 드론 시리얼 번호 식별 결과이다. 해당 드론 조종 어플리케이션을 통해 드론을 비행한 경우 드론 기체의 'PS'로 시작하는 시리얼 번호의 하위 10글자가, 드론 조종 어플리케이션 데이터에 기록된 'PI'로 시작하는 시리얼 번호의 하위 10글자와 동일한 값을 갖는다. 이러한 방식의 비교를 통해 상관성 분석 대상 조종자 계정 또는 조종기에서 특정 드론의 비행을 수행했는지 여부를 확인할 수 있다.



[그림 4-4] Parrot 드론 환경에서의 상관성 분석

제 2 절 6W 관점의 드론 범죄 재구성

드론 포렌식 과정에서 드론 범죄의 재구성은 관련 디지털 증거 간의 연계 분석 효율성을 향상시킨다. 본 연구에서는 드론 조종기와 클라우드에서 획득한 데이터의 분석을 통해 식별 가능한 정보를 6W (Who, Where, Which, What, When, Where to Where) 관점에서 분류하여 드론 범죄를 재구성한다. [표 4-1]는 드론 조종기 환경별 식별 가능한 정보에 대해 6W 관점에서 분류한 결과이다. 이러한 분류를 통해 모든 조종 환경에서

6W를 식별할 수 있는 데이터를 획득할 수 있음을 확인하였다.

[표 4-1] 6W 관점에서 분석 데이터의 분류 결과

6W	Data	Drone controller
Who	Profile image	DJI GO·GO4·Fly·Pilot·SC
	Email	DJI GO·GO4·Fly·Pilot·SC, Parrot Freeflight 6
	RC S/N	DJI Fly·SC, Parrot Freeflight 6
Where	Real-time pilot location	Parrot Freeflight 6
	Drone home point	DJI GO·GO4·Fly·Pilot·SC
Which	Drone serial number	DJI GO·GO4·Fly·Pilot·SC, Parrot Freeflight 6
What	Drone flight path	DJI GO·GO4·Fly·Pilot·SC, Parrot Freeflight 6
	Media files captured by the drone	DJI GO·GO4·Fly·Pilot·SC, Parrot Freeflight 6
When	Drone flight time	DJI GO·GO4·Fly·Pilot·SC, Parrot Freeflight 6
	Media capture timestamp of the drone	DJI GO·GO4·Fly·Pilot·SC, Parrot Freeflight 6
Where to Where	Drone flight path	DJI GO·GO4·Fly·Pilot·SC, Parrot Freeflight 6

Who는 조종자의 정보를 의미한다. 조종자의 정보 중 프로필 이미지는 드론 조종 어플리케이션에서 조종자가 설정한 프로필 이미지를 의미한다. 모든 DJI 드론 조종 환경에서는 이러한 조종자의 프로필 이미지를 획득할 수 있다. 이메일은 조종자의 드론 조종 어플리케이션의 계정 정보를 의미하며, 모든 드론 조종 환경에서 획득할 수 있다. 조종자가 드론 어플리케이션을 특정 조종기에서 구동한 경우, 어플리케이션 데이터에 이러한 조종기의 시리얼 번호가 기록된다. 본 연구에서는 DJI Fly 어플리케이션 및 DJI SC, Parrot Freeflight 6 어플리케이션에서 이러한 조종기의 시리얼 번호를 식별할 수 있음을 확인하였다.

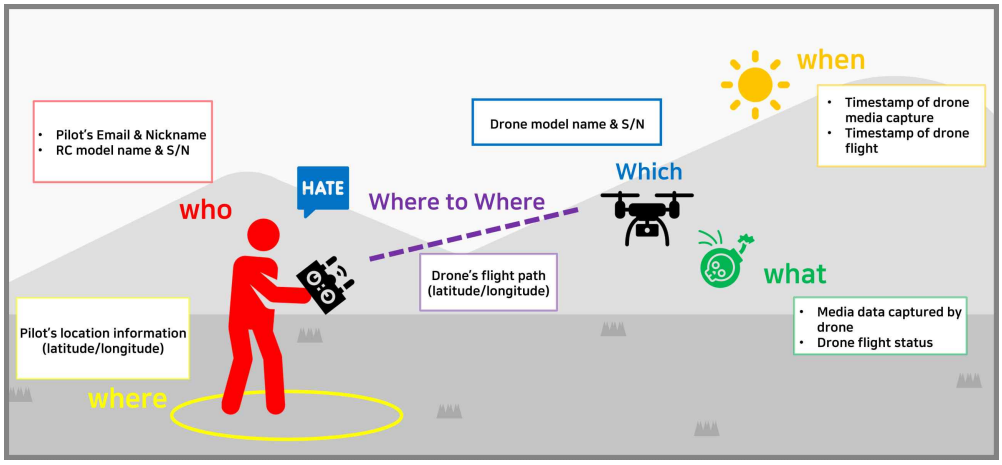
Where은 조종자의 위치를 의미한다. 이러한 정보는 드론의 비행을 통해 생성된 비행로그 파일을 통해 획득할 수 있다. DJI 드론 환경에서는 드론의 귀환 위치인 홈포인트가 비행로그 파일에 기록된다. 이는 드론이 비행을 시작 시 자동적으로 생성되며, 조종자가 임의로 홈포인트를 변경할 수 있다. 조종자는 RTH(Return to Home) 기능을 통해 자동으로 드론을 홈포인트로 착륙시킬 수 있다. 드론 조종자는 드론의 회수를 위해 조종자가 물리적으로 접근 가능한 범위 내에서 홈포인트를 설정을 하며, 비행로그 파일에 기록된 홈포인트는 조종자가 물리적으로 이동 가능한 위치를 의미한다. 따라서 본 연구에서는 DJI 드론 환경에서의 홈포인트를 조종자의 위치로 해석하였다. Parrot의 비행로그 파일에는 드론이 비행 중인 상황에서 조종자의 위치를 실시간으로 기록한다. 이러한 정보는 DJI 환경과는 다르게 직접적으로 조종자의 이동 경로 및 비행의 시작과 종료 시점에서의 위치를 식별하는 단서를 제공한다.

Which는 비행을 수행한 드론을 의미한다. 본 연구에서는 모든 드론 조종 환경에서 비행을 수행한 드론의 시리얼 번호가 저장되는 것을 확인하였으며, 이를 실제 드론 기체에 표기된 시리얼 번호와 비교하여 상관성 증명 방법을 식별하였다.

What과 When은 각각 드론과 조종자의 행위 및 행위 시각을 의미한다. 드론 범죄를 통해 발생할 수 있는 대표적인 행위는 드론의 비행과 미디어 촬영 행위이다. 본 연구에서는 드론 조종 환경에서 생성된 비행로그 파일을 통해 드론의 비행 경로 및 시간 정보를 획득하였으며, 미디어 파일과 미디어 파일의 메타데이터 및 파일명 등을 통해 촬영 시각을 식별하였다.

Where to Where은 드론의 비행 경로이다. 드론 포렌식 관점에서 드론의 비행 경로는 중요한 단서이며, 본 연구에서는 비행로그 파일에 기록된 드론의 실시간 이동 경로를 식별하였다.

이러한 6W 관점에서의 데이터 분류를 통해 결과적으로 [그림 4-5]와 같이 드론 포렌식 통합 분석을 위해 6W 요소에 따른 드론 조종 환경별 획득 가능한 데이터 범위를 식별할 수 있다.



[그림 4-5] 6W 관점에서의 드론 포렌식 통합 분석 결과

HANSUNG UNIVERSITY

제 5 장 결론

드론 기술의 발전과 함께 드론은 민간 분야를 비롯하여 국방, 물류, 농업 등 다양한 산업 분야에서 폭넓게 활용되고 있다. 드론의 원격 비행 및 미디어 촬영 기능은 기존에 인력이 직접 수행하기 어려웠던 작업을 효율적으로 수행할 수 있도록 하며, 비교적 낮은 진입 장벽으로 누구나 쉽게 장비를 구매하고 운용할 수 있다는 특징을 가진다. 그러나 이러한 장점과 함께 드론을 악용한 불법 비행, 불법 촬영, 물품 운반 등 다양한 범죄 사례 또한 지속적으로 증가하고 있다. 이러한 환경에서 드론 관련 사건의 실체를 규명하기 위한 디지털 포렌식 기술의 중요성은 점차 커지고 있다. 기존의 드론 포렌식 연구는 주로 드론 기체에 저장된 데이터 분석을 중심으로 수행되어 왔으며, 단일 조종 어플리케이션 또는 조종기를 대상으로 한 제한적인 분석에 머무르는 한계가 있었다.

본 연구에서는 드론 조종자를 중심으로 한 디지털 증거 수집 및 분석 기법을 제안하였다. 이를 위해 모바일 환경과 임베디드 조종기 환경에서 동작하는 드론 조종 어플리케이션을 대상으로 로컬 데이터를 수집하고, 저장된 암호화 데이터를 복호화하여 디지털 증거로 활용하였다. 또한 기존 연구에서 충분히 다루어지지 않았던 드론 조종 어플리케이션 기반의 클라우드 포렌식 분석을 수행하여, 드론 제조사의 원격 서버에 저장된 조종자 데이터를 수집하였다. 수집된 조종기 로컬 데이터와 클라우드 데이터를 대상으로 통합 분석을 수행하고, 이를 6W 관점에서 체계적으로 분류함으로써 드론 범죄 재구성을 위한 분석 절차를 제시하였다. 그 결과, 다양한 환경으로부터 확보된 디지털 증거를 종합하여 드론 범죄 사건의 전반적인 행위 흐름을 재구성할 수 있음을 확인하였다.

본 연구의 결과는 드론 기체 중심의 기존 포렌식 분석 범위를 확장하여, 조종자 중심의 디지털 증거 수집과 분석을 통한 드론 범죄 재구성의 가능성을 제시한다. 이를 통해 향후 드론 관련 범죄 수사에서 드론의 비행 경로뿐만 아니라 조종자의 행위 및 책임 소재를 보다 정밀하게 추적·규명하는 데 기여할 수 있을 것으로 기대된다.

참 고 문 헌

1. 국내문헌

한겨레신문, “몰카·절도·마약 밀반입... ‘드론 범죄’ 급증”, Available online: https://www.hani.co.kr/arti/international/europe/755665.html (accessed on 05 July 2025).

서울신문, “‘불법 드론’ 중국인 유학생들 구속... 부산 입항 美 항모 등 촬영”, Available online: https://www.seoul.co.kr/news/society/accident/2025/06/26/20250626500111, (accessed on 05 July 2025).

KBS 뉴스, “고층 아파트 또 드론 ‘몰카’... 수사도 못해”, Available online: https://news.kbs.co.kr/news/pc/view/view.do?ncd=8121561 (accessed on 05 July 2025).

정보통신산업진흥원, “품목별 ICT 시장 동향”, Available online: https://www.globalict.kr/upload_file/kms/202312/46257328044971008.pdf (accessed on 05 July 2025).

Youngwoo Lee, Juhwan Kim, Jihyeon Yu and Joobeom Yun, "Classification of DJI Drones Based on Flight Log Decryption Method", Journal of The Korea Institute of Information Security & Cryptology, 32(1), pp.77-88, 2022.

YeoHoon Yoon and Joobeom Yun, "Analysis of the Possibility of Recovering Deleted Flight Records by DJI Drone Model", Journal of The Korea Institute of Information Security & Cryptology, 33(4), pp.609-619, 2023.

Dong-Min Woo and Young-Ho Park, "Drone Log Security Using PX4", Journal of The Korea Institute of Information Security & Cryptology, 34(6), pp.1239-1250, 2024.

Sukyung Cho, Seonghyeon Lee, Mingyu Seong, Subin Jo and Jungheum Park, "A Study on Categorizing and Analyzing Flight Logs generated

by DJI Drones", Journal of Digital Forensics, 17(1), pp.13-25, 2023.
Sehoon Lee, Seongyeol Oh, Seungoh Seo and Junyoung Son, "A Study on Acquiring Data from DJI Drone using DUML Protocols", Journal of Digital Forensics, 17(2), pp.89-103, 2023.
Seyoung Baik and Sangwook Lee, "A Study On Optimized Drone Forensic Methodology Applied with Open Source Based Drone Live Forensic Tool", Journal of The Korea Institute of Information Security & Cryptology, 33(4), pp.633-646, 2023.

2. 국외문헌

Federal Aviation Administration, "Drone Sightings Near Airports", Available online: https://www.faa.gov/uas/resources/public_records/uas_sightings_report (accessed on 05 July 2025).
Devon R. Clark, Christopher Meffert, Ibrahim Baggili and Frank Breitinger, "DROP (DRone Open source Parser) your drone: Forensic analysis of the DJI Phantom III", Digital Investigation, 22, 2017.
Ankit Renduchintala, Farha Jahan, Raghav Khanna and Ahmad Y. Javaid, "A comprehensive micro unmanned aerial vehicle (UAV/Drone) forensic framework", Digital Investigation, 30, pp.52-72, 2019.
Yousef M., Iqbal F. and Hussain M., "Drone forensics: a detailed analysis of emerging DJI models", Proceedings of the International Conference on Information and Communication Systems, pp.45-50, 2020.
Chi-Cheng Yang, Hsuan Chuang and Da-Yu Kao, "Drone Forensic Analysis Using Relational Flight Data: A Case Study of DJI Spark and Mavic Air", Procedia Computer Science, 192, pp.1359-1368, 2021.

Khalifa Al-Room, Farkhund Iqbal, Thar Baker, Babar Shah, Benjamin Yankson, Aine MacDermott and Patrick C. K. Hung, "Drone Forensics: A Case Study of Digital Forensic Investigations Conducted on Common Drone Models", International Journal of Digital Crime and Forensics, 13(1), 2021.

Fahad E. Salamh, Mohammad Meraj Mirz and Umit Karabiyik, "UAV Forensic Analysis and Software Tools Assessment: DJI Phantom 4 and Matrice 210 as Case Studies", Electronics, 10(6), 2021.

Hyunji Moon, Euihyun Jin, Hyeon Kwon, Sangeun Lee and Gibum Kim, "Digital forensic methodology for detection of abnormal flight of drones", Journal of Information Security and Cybercrime Research, 4(1), 2021.

Stankovic M., Mirza M. and Karabiyik U., "UAV forensics: DJI Mini 2 case study", Drones, 5(2), 2021.

Husnjak S., Forenbacher I., Perakovic D. and Cvitic I., "UAV forensics: DJI Mavic Air noninvasive data extraction and analysis", Proceedings of the 5th EAI International Conference on Management of Manufacturing Systems, pp.123-130, 2022.

Sungwon Lee, Hyeongmin Seo and Dohyun Kim, "Digital forensic research for analyzing drone pilot: Focusing on DJI remote controller", Sensors, 23(21), 2023.

Ziyu Zhao, Yongquan Wang and Genwei Liao, "Digital forensic research for analyzing drone and mobile device: Focusing on DJI Mavic 2 Pro", Drones, 8(7), 2024.

Hana Bouafif, Faouzi Kamoun and Farkhund Iqbal, "Towards a Better Understanding of Drone Forensics: A Case Study of Parrot AR Drone 2.0", International Journal of Digital Crime and Forensics, 12(1), 2020.

Kumar R. and Agrawal A., "Drone GPS data analysis for flight path

reconstruction: a study on DJI, Parrot, Yuneec make drones", Digital Investigation, 38, 2021.

I CAN Hack, "Local Privilege Escalation on the DJI RM500 Smart Controller", Available online: https://icanhack.nl/blog/dji-rm500-privilege-escalation (accessed on 05 July 2025).

Android Developers, "Android Debug Bridge (adb)", Available online: https://developer.android.com/tools/adb (accessed on 05 July 2025).

Github, "frida", Available online: https://github.com/frida/frida (accessed on 05 July 2025).

Github, "MemDumper: Dump Memory of Process in Android", Available online: https://github.com/kp7742/MemDumper (accessed on 05 July 2025).

DJI, "DJI Developer", Available online: https://developer.dji.com/ (accessed on 05 July 2025).

Github, "FlightRecordParsingLib", Available online: https://github.com/dji-sdk/FlightRecordParsingLib (accessed on 05 July 2025).

AirData, "Drone Data Management and Flight Analysis | Airdata UAV", Available online: https://airdata.com (accessed on 05 July 2025).

ABSTRACT

A Study on Drone Forensics through Drone Controller and Cloud Data Analysis

Kim, Hangyeol

Major in Convergence Security

Dept. of Convergence Security

The Graduate School

Hansung University

As the use of drones continues to expand across both civilian and industrial sectors, drones provide various operational benefits while also being increasingly exploited as instruments for criminal activities. Owing to the characteristic that operators can perform actions remotely and be physically separated from the scene, forensic analysis based on digital evidence—such as flight logs and media files—is essential in drone-related crimes. Although existing drone forensic studies have primarily focused on analyzing data stored on the drone itself, operator-centric digital evidence analysis aimed at identifying operators and tracing their activities is crucial in real investigative environments. In particular, in modern drone ecosystems where operator data are managed through cloud-based services, forensic analysis of remotely stored data on manufacturers' servers is required. Accordingly, this study proposes an operator-centric digital forensic analysis methodology targeting DJI and Parrot drones, which hold a significant share of the global drone market, by incorporating both mobile control applications and embedded remote controllers. By collecting and analyzing operator-related data from the local storage of controllers and cloud servers, this study presents an integrated analysis procedure for reconstructing drone-related crimes.

【Key words】 Digital Forensics, Drone Forensics, DJI Drone, Parrot Drone