UAV 의 통신을 위한 MAVLink 취약점 분석과 대응 방안

황연정, 강은세, 신다윗, *김호원
부산대학교
yeonjeong@islab.re.kr, eunse@islab.re.kr, dawit@islab.re.kr, *howonkim@gmail.com

Analysis of MAVLink Vulnerabilities for UAV Communications and Countermeasures

Yeonjeong Hwang, Eunse Kang, Dawit Shin, *Howon Kim
Pusan National University

요 약

MAVLink 는 성능 오버헤드가 낮으며 시스템간 호환성이 뛰어나 UAV 와 GCS, 로봇 사이에 가장 널리 쓰이는 프로토콜 중 하나이다. 하지만 MAVLink 에 대한 보안 연구는 아직 미흡하여 많은 취약점이 존재한다. 이에 대비하여 본 연구에서는 MAVLink 를 사용하는 UAV 에 대해 공격 시나리오를 가정하고 시뮬레이션으로 검증했으며 취약점에 대한 대응 방안을 제시하였다.

Ⅰ. 서론

UAV(Unmanned Aerial Vehicle)는 지상에 있는 GCS 의 신호에 따라서 자동/반자동으로 날아가는 무인 항공기이며 기술적 발전을 거듭하여 군용 목적, 농업, 지리적 데이터 수집 등 다양한 분야에 폭넓게 활용되고 있다. 이에 따라 UAV 의 통신 또한 발전하고 있으며 MAVLink 는 UAV, GCS(Ground Control Station)와 로봇 사이의 통신으로 가장 많이 사용되고 있는 프로토콜 중 하나이다.

MAVLink 는 UAV 와 같은 경량 환경에서 주로 사용되어 통신 데이터 패킷이 짧은 특징을 가진다. 따라서 한번에 전송할 수 있는 데이터 길이가 한정적이며 이로 인해 암복호화 및 인증 메커니즘을 제공하지 않아 기밀성, 무결성에 대한 취약점이 존재한다. MAVLink 1.0 의 경우 인증 메커니즘이 존재하지 않아 중간자가 GCS Spoofing 기법을 통해 UAV 를 탈취할 수 있으며[1], 해당 취약점을 보완하기 위해 서명 기반 인증을 제공하는 MAVLink 2.0 이 출시되었으나 2.0 또한 UAV 의 주요 정보를 평문으로 전달하는 취약점이 존재한다. 해당 취약점은 스니핑 공격에 취약한 무선 통신 환경의 특성상[3] GPS Spoofing 과 같은 다른 공격으로 이어질 소지가 있다.

이에 대비하여 본 논문에서는 시뮬레이션 상의 무선 환경에서 MAVLink 를 사용하는 UAV 에 대하여 공격 시나리오를 가정하고 피해로 이어질 수 있는지 추산한다. 또한 취약점을 보완하여 기밀성과 무결성을 제공하는 대응 방안을 제시한다.

Ⅱ. 본론

2.1 시뮬레이션 환경

논문에서 제시하는 MAVLink 의 취약점을 확인하기 위해 시뮬레이션 환경을 구성했으며 Linux 가상 환경에서 PX4 와 Gazebo Simulator 를 사용한다. PX4 는 XRCE-DDS 에이전트로 구성되어 DDS(Data Distribution Service) 통신을 지원하고 있으며, PX4 와 ROS2 노드 간의 통신을 위해 ROS2 foxy 를 사용하여 XRCE-DDS 미들웨어를 설정했다. GCS 는 Qground Control 오픈소스를 사용하여 구현했으며, GCS 가 로컬과 외부에서 각각 PX4 와 연결되어 동시 접속한다. 구현한 로컬 시뮬레이션 환경의 구성도는 아래 그림과 같다.

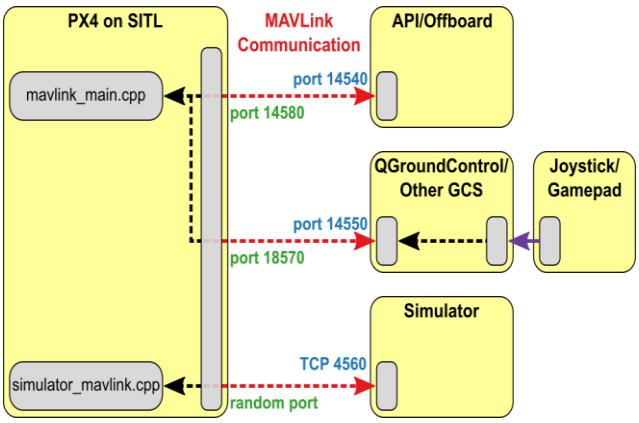

그림 1. 구성 로컬 시뮬레이션 환경

2.2 공격 시나리오 및 취약점 분석

MAVLink 1.0 은 암호화를 통한 기밀성과 무결성 인증을 제공하지 않는다. MAVLink 2.0 에는 프로토콜 패킷의 헤더부에 signature 를 추가하여 인증 메커니즘을 도입하였으나, 1.0 버전과 마찬가지로 기밀성을 제공하지 않는다.

따라서 공격자가 GCS-UAV 통신에 사용하는 무선 네트워크 내에 위치하는 경우, 공격자는 MAC Spoofing 을 통한 패킷 스니핑으로 GCS 와 UAV 가 MAVLink 프로토

콜을 통해 주고 받는 메시지를 평문으로 볼 수 있다. 공격자는 MAVLink 메시지로 전달되는 UAV 의 실시간 정보를 이용하여 GPS Spoofing 과 같은 공격을 시도할 수 있다. 아래 그림은 공격자의 서버에서 스니핑한 GCS-UAV 의 MAVLink 패킷이다.

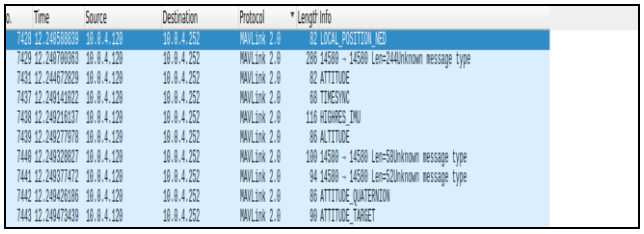

그림 2. 스니핑된 MAVLink 패킷

또한 MAVLink 2.0 은 서명을 통한 인증 기능을 제공하고 있으나 AUTOPILOT_MESSAGE 를 통해 강제로 버전을 다운그레이드 하는 취약점이 있다.[4] 그것에 더해, 구성한 시뮬레이션 환경과 같이 하나의 UAV 에 다수의 GCS(Qground Control)가 연결되어 있는 경우, GCS 가 MAVLink 2.0 버전을 1.0 으로 바꾸는 동시에 다른 GCS 의 MAVLink 버전이 변경 됨을 확인 가능하다. 이는 취약점이 존재하는 포인트에서 GCS Spoofing 공격을 성공적으로 수행하면 노드 전체가 기밀성 및 무결성을 보장하지 않는 MAVLink 1.0 으로 통신하게 됨을 의미한다.

아울러, MAVLink 에서는 메시지의 무결성 인증을 위한 checksum 을 계산하기 위해 CRC-16 을 이용하나 키를 사용하지 않고 선형적인 구조를 가지므로 공격자가 기존 프레임을 수정하면서 CRC 값을 쉽게 업데이트 할 수 있다. 이는 공격자가 위조된 프레임을 주입하는 경우 CRC 를 독립적으로 계산하여 추가할 수 있으므로 MAVLink 의 무결성을 완벽하게 보장하지 못한다.[5]

2.3 취약점 대응 방안

MAVLink 는 암호화를 통한 기밀성을 제공하지 않아 MITM 공격에 취약하며, 공격자가 사후 공격에 이용할 수 있는 실시간 정보가 평문으로 노출된다. 또한 signature 유무가 MAVLink 의 버전에 좌우되므로 공격자가 MAVLink 의 backward compatibility 를 이용하여 signature 를 제거할 소지가 있다. 따라서 기밀성과 무결성을 보장하는 암호 알고리즘이 요구되며 AES, ARIA 등을 사용할 수 있다.

표준 암호 알고리즘으로 사용되고 있는 AES-CCM(Counter with CBC-MAC), AES-GCM(Galois Counter Mode)의 경우 CTR 모드로 암호화하여 스트림 암호로 사용이 가능하다. AES-CCM 의 경우 메세지와 AAD(Additional Authenticated Data)길이에 제한이 있기 때문에 가변 길이를 사용하는 MAVLink 패킷 구조에 적합하지 않다.

AES-GCM 은 넓은 범위의 메세지 크기와 AAD 크기를 지원하여 패킷 구조에 적합하며, 인증 태그를 계산할때 GHASH 함수를 사용해 무결성을 제공한다. 그러나 SPN(Substitution Permutation Network)구조로 인해 암복호화시 연산 오버헤드가 크기 때문에 경량 환경에 적용하기 어렵다.[6,7] 반면 ARIA-GCM 은 AES-GCM 과 비교하여 키 크기는 동일하나 Involutional SPN 구조를 사용한다. 따라서 데이터의 암호화 및 복호화에 다른 연산을 구현할 필요가 없으므로 경량 환경에 적합한 구조를 가진다.

이에 본 논문에서는 MAVLink 의 취약점을 보안하기 위해 기밀성 및 무결성을 동시에 제공하며 경량 환경에서 사용하기 적합한 암호 알고리즘인 ARIA-GCM 을 제안한다.

일반적으로 암호알고리즘을 하드웨어 구현했을 시에 소프트웨어 구현에 비해 월등히 빠른 처리 속도와 낮은 전력 소모량을 보인다.[8] 같은 키 사이즈의 AES 와 ARIA 를 비교했을 때, 소프트웨어 구현 시에는 ARIA 가 더 많은 라운드 수를 가지므로 처리량이 더 낮다. 하지만 하드웨어 구현 시에 메모리가 제한된 환경에 적합한 on-the-fly 방식으로 키 스케줄러를 구현할 경우 ARIA 가 AES 보다 빠른 복호화 속도를 보인다. [9] 그 외에도 ARIA-GCM 의 하드웨어 구현 시 파이프라인을 통한 병렬 처리와 같은 가속 기법을 적용하고, UAV 의 저사양 환경에 맞게 최적화한 연산을 구현하여 성능과 처리량을 더 높일 수 있다.[10,11]

Ⅲ. 결론

본 논문에서는 UAV 의 통신 프로토콜인 MAVLink 의 취약점을 분석하고 공격 시나리오를 가정하여 시뮬레이션으로 검증하였다. 시나리오를 통해 MAVLink 의 기밀성과 무결성에 대한 취약점을 발견했으며 이를 바탕으로 기밀성과 무결성을 제공하는 ARIA-GCM 을 제시하였다. 하지만 암호 알고리즘의 키 관리에 대한 취약점은 여전히 존재하며 대응 방안이 존재하지 않아 이에 대해 연구될 필요성이 있다.

또한 다양한 보안 솔루션이 제시됨에 따라 성능 오버헤드가 증가하게 되고 실시간성을 보장하는데 어려움이 있다. 이는 ARIA-GCM 에 병렬 처리를 적용하는 하드웨어 가속기가 구현될 필요성을 시사한다. 하드웨어 가속기를 이용한 암호 알고리즘의 암복호화 구현시 동일한 연산을 사용하는 ARIA 를 보다 효율적으로 사용할 수 있을 것으로 기대된다.

ACKNOWLEDGMENT

“본 연구는 과학기술정보통신부 및 정보통신기획평가원의 융합보안핵심인재양성사업의 연구 결과로 수행되었음” (IITP-2023-2022-0-01201)

참 고 문 헌

[1] Deligne, E. ARDrone corruption. J Comput Virol 8, 15–27 (2012). https://doi.org/10.1007/s11416-011-0158-4

[2]S.Kamkar, ”Skyjack,“https://github.com/samyk/skyjack,Dec.

[3] Min-kyu Choi, Rosslin John Robles, Tai-hoon Kim, Chang-hwa Hong. (2008). Wireless Network Security: Vulnerabilities, Threats and Countermeasures. International Journal of Multimedia and Ubiquitous Engineering, 3(3), 77-86.

[4] X. Yang, E. Karampatzakis, C. Doerr and F. Kuipers, "Security Vulnerabilities in LoRaWAN," 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), Orlando, FL, USA, 2018, pp. 129-140, doi: 10.1109/IoTDI.2018.00022.

[5] A. Allouch, O. Cheikhrouhou, A. Koubâa, M. Khalgui and T. Abbes, "MAVSec: Securing the MAVLink Protocol for

Ardupilot/PX4 Unmanned Aerial Systems," 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 2019, pp. 621-628, doi: 10.1109/IWCMC.2019.8766667.

[6] N. Prapulla, S. Veena and G. Srinivasalu, "Development of algorithms for MAV security," 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 2016, pp. 799-802, doi: 10.1109/RTEICT.2016.7807936.

[7] A. Allouch, O. Cheikhrouhou, A. Koubâa, M. Khalgui and T. Abbes, "MAVSec: Securing the MAVLink Protocol for Ardupilot/PX4 Unmanned Aerial Systems," 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 2019, pp. 621-628, doi: 10.1109/IWCMC.2019.8766667.

[8] Isil Çetintav, Deniz Taskin; Performance analysis of hardware and software based AES encryption on internet of things SoC. AIP Conf. Proc. 1 September 2023; 2849 (1): 190004. https://doi.org/10.1063/5.0163518.

[9] Sang Woo Lee, Sang-Jae Moon, Jeong-Nyeo Kim. (2008). High-Speed Hardware Architectures for ARIA with Composite Field Arithmetic and Area-Throughput Trade-Offs. ETRI ETRI Journal, 30(5), 0-0.

[10] Heung-Ryol Yoo, Sun-Jong Lee, & Yung-Deug Son (2018). Hardware Design and Implementation of Block Encryption Algorithm ARIA for High Throughput. Journal of IKEEE, 22(1), 104-109.

[11] Seo, Hwajeong, Hyeokdong Kwon, Hyunji Kim, and Jaehoon Park. 2020. "ACE: ARIA-CTR Encryption for Low-End Embedded Processors" Sensors 20, no. 13: 3788. https://doi.org/10.3390/s20133788.