Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.5, Oct. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.5.737

보안 UAV를 위한 암호모듈의 성능평가와 보안성 평가 방법에 대한 연구

김 용 대,†‡ 김 덕 진, 이 은 경, 이 상 욱
ETRI 부설연구소 (연구원)

A Study On Performance Evaluation of Cryptographic Module and Security Functional Requirements of Secure UAV

Yongdae Kim,†‡ Deokjin Kim, Eunkyoung Yi, Sangwook Lee
The Attached Institute of ETRI (Researcher)

요 약

무인 항공기 (Unmanned Aerial Vehicle, UAV)는 4차 산업혁명 시대와 함께 매우 빠르게 성장하고 있다. 또한, 인공지능 기술과 반도체 기술의 발전에 따라 무인 항공기의 성능이 향상됨에 따라서 기존의 군용 등 특수 목적으로 사용되던 것에서 현재는 취미, 교량 점검, 정찰 등 다양한 민간 분야에도 활용되기 시작했다. 현재 민간 분야의 무인 항공기에서 가장 많이 활용되는 통신 프로토콜은 오픈 소스로 시작한 MAVLink(Macro Air Vehicle)이다. 하지만 MAVLink에는 보안 기술이나 암복호화 메커니즘은 포함되지 않아 현재 무인 항공기의 보안 위협에 취약할 수 밖에 없다. 따라서, 본 연구에서는 기밀성을 보장하기 위해서 암호모듈을 무인 항공기에 구현하고 다양한 구현 방식에 따른 무인 항공기에서의 암호화 및 복호화 성능 평가에 대한 결과를 분석한다. 또한, 보안 모듈을 적용한 무인 항공기에 위협 분석을 통하여 보안요구사항을 국제 표준 Common Criteria에 따라 평가 기준에 대해 논한다.

ABSTRACT

The demands of Unmanned Aerial Vehicles (UAVs) are growing very rapidly with the era of the 4th industrial revolution. As the technology of the UAV improved with the development of artificial intelligence and semiconductor technology, it began to be used in various civilian fields such as hobbies, bridge inspections, etc from being used for special purposes such as military use. MAVLink (Macro Air Vehicle Link), which started as an open source project, is the most widely used communication protocol between UAV and ground control station. However, MAVLink does not include any security features such as encryption/decryption mechanism, so it is vulnerable to various security threats. Therefore, in this study, the block cipher is implemented in UAV to ensure confidentiality, and the results of the encryption and decryption performance evaluation in the UAV according to various implementation methods are analyzed. In addition, we proposed the security requirements in accordance with Common Criteria, which is an international recognized ISO standard.

Keywords: Block Cipher, UAV, Common Criteria, MAVLink

Received(04. 27. 2022), Modified(1st: 07. 22. 2022, 2nd: 08. 29. 2022), Accepted(08. 29. 2022)

† 주저자, kimyd@nsr.re.kr
‡ 교신저자, kimyd@nsr.re.kr(Corresponding author)

I. 서 론

무인 항공기(UAV)는 원격 제어를 통해서 사람이 항공기 내부에서 조정하지 않고 무인으로 항공기를 운행하거나 또는 완전 자율로 운행하는 항공기를 말한다. 무인 항공기 이외에도 운행하는 위치 및 환경 등에 따라서 육상에서 이용되는 육상무인이동체(Unmanned Ground Vehicle), 해양무인이동체(Unmanned Maritime Vehicle) 등이 있다.

기존의 무인 항공기는 군용 등 특수한 목적의 무인 항공기가 대부분을 차지하였으나 최근 들어 군수분야 이외에도 매년 그 수요는 증가하고 있다. 군용으로는 무인 정찰기 및 폭격기 등으로 사용되었으나 민간에서 취미, 농업 지원, 교량 등의 측정 등을 위한 목적, 산림 보호 및 감시, 시설물 안전 검사, 순찰 등의 무인 항공기의 수요가 증가하고 있다. 민간용 무인 항공기의 시장은 2018년까지 연평균 30% 이상의 높은 성장을 보여 2026년에는 약 118억 달러에 이를 것으로 추정되고 있다.

이러한 민간용 무인 항공기에서 MAVLink(Macro Air Vehicle Link)라는 통신 프로토콜이 가장 많이 활용되고 있다 [1]. 현재 MAVLink는 2.0 버전까지 나왔지만 통신 패킷의 무결성 검증을 포함한 것 이외에 데이터의 기밀성을 보장하기 위한 암복호화같은 메커니즘은 포함하지 않고 있다 [2]. 무인 항공기의 메인 칩의 한정된 리소스를 이유로 시작부터 MAVLink는 경량화를 중점을 두면서 시작되어 보다 많은 리소스를 필요로 하는 보안 기술이나 암복호화는 고려되지 않았다. 이러한 이유로 MAVLink 프로토콜을 활용한 통신에서의 데이터 메시지 위변조 및 서비스 거부(Denial Of Service), 스푸핑(Spoofing) 등의 다양한 통신 취약점에 노출되게 된다 [3], [4].

지금까지 무인 항공기의 보안 취약점에 대한 다양한 연구가 진행되어 왔다 [5], [6], [7], [8]. 2015년 USENIX에서는 드론에 탐재된 자이로 센서에 직접적으로 공격을 하여 오작동을 유발시켜 드론을 탈취 및 추락시킬 수 있는 취약점을 발견하였다 [9]. 드론을 컨트롤 하는 것 이외에 드론에 가짜 GPS 신호를 전송하는 GPS Spoofing 취약점을 있음을 발견하였다 [10], [11]. 이러한 방식의 공격 이외에 통신 채널인 MAVLink 프로토콜의 취약점을 이용한 다양한 공격이 가능함을 2018년의 논문에서 소개되었다 [12]. 저자는 실험을 통해서 지상 관제 센터(Ground Control Station)과 무인 항공기(UAV)와의 통신 채널에서 악의적인 패킷을 입력하여 UAV가 비정상적인 동작을 하는 것을 보였다. 또한 퍼징(Fuzzing) 기법을 통해서 MAVLink의 통신 채널에서의 취약점을 찾아낼 수 있음을 발견하였다 [13]. 실제 본 논문에서는 Floating Point Exception을 일으킬 수 있는 소프트웨어적인 취약점을 발견하였다.

이러한 보안 위협을 감소시키기 위해서 MAVLink에 다양한 블록암호를 적용한 MAVLink를 제안하였다 [14]. 해당 논문에서 사용한 알고리즘은 AES-CTR, AES-CBC, ChaCha2, RC4가 있다. 하지만 해당 논문에서는 실제 드론에 구현한 것이 아닌 시뮬레이션 환경(Software In The Loop, SITL)에서 측정한 것으로 실제 드론에서 적용 가능성 및 성능에 대해서는 제시하지 않고 있다.

또한, 2021년에는 MAVLink의 네트워크 레벨의 침입 탐지 시스템(Intrusion Detection System, IDS)을 제안하였다 [15]. 저자는 실제 DDoS 공격의 일종인 Flooding 공격을 MAVLink 상에서 수행했을 때에 제안된 IDS로 탐지됨을 보였다. 하지만, 이는 네트워크 통신 패킷에 대한 탐지 방법에 대한 것으로 통신 패킷의 기밀성을 적용하기 위한 제안 방식은 아니다.

논문 [16]의 저자는 MAVLink 통신 채널의 기밀성을 위해서 OTP(One-Time Pad) 암호화 방식을 적용하였다. 저자는 3DES, Twofish, AES 등과 같은 블록암호를 사용했을 때 비해서 높은 처리속도와 데이터 정확도를 나타내였다. 그러나 구체적으로 OTP 암호화 방식을 사용하기 위해 필요한 Pad 값을 어떻게 UAV와 지상관제센터가 공유가 되는지에 대한 설명이 없고, 실 기기에 적용한 테스트 결과인지가 불분명하다.

본 논문에서는 데이터 기밀성을 유지하기 위해 다양한 블록암호 알고리즘을 SW 형태로 구현하였다. 구현된 암호모듈을 실제 무인 항공기의 플랫폼인 PX4 [17]에 탑재하여 어느 정도의 리소스가 추가적으로 필요한지 각 블록암호 알고리즘 및 구현 방식에 따라 평가하였다. 본 연구에서는 AES, SEED, ARIA, LEA의 4종의 블록암호 알고리즘을 ECB 및 CBC 운영 모드로 SW 암호모듈을 구현하였다. 키 사이즈는 모두 128비트를 이용하였다.

또한, 각 알고리즘 및 운영 모드에 따라 메모리 및 속도 최적화 방식의 구현 방법을 적용하여 각각의 구현 방식에서의 결과를 비교 분석하였다. 마지막으로는 이러한 암호모듈이 적용된 무인 항공기의 안전성 평가를 위한 국제공통평가기준(Common Criteria)를 통한 보안요구사항을 제시함으로써 무인 항공기 암호모듈의 공통평가 기준에 대해 논하고자 한다.

II. 암호모듈 구현 방법

2.1 암호 알고리즘

2.1.1 SEED

1999년 2월 한국인터넷진흥원에서 개발한 대칭키 블록암호 알고리즘이다. 2005년에는 국제 표준화 기구인 ISO/IEC 국제블록암호알고리즘 IETF 표준으로 제정되었다. 국내는 1999년 TTA의 128비트 블록암호알고리즘(SEED)으로 제정되었다. 그 이외에 SEED 암호 알고리즘 자체에 대한 표준 외에도 SEED를 사용하기 위한 다양한 국내외 표준이 제정되었다 [18].

기본적으로 SEED 암호알고리즘은 DES와 동일하게 Feistel 구조로 이루어졌다. 128비트 블록 입력으로 총 16라운드로 구성되었다. 내부 함수에서의 연산은 XOR, ADD만 사용되며 2개의 안전성이 입증된 S-Box를 사용한다. G함수 내부에서 사용되는 비선형 S-Box는 다음의 식을 이용하여 성립된다.

$$S_i : Z_{2^8} \rightarrow Z_{2^8},\ \mathrm{S(x)} = \mathrm{A}^{(i)} \bullet \mathrm{x}^{n_i} \oplus \mathrm{b_i} \qquad (1)$$

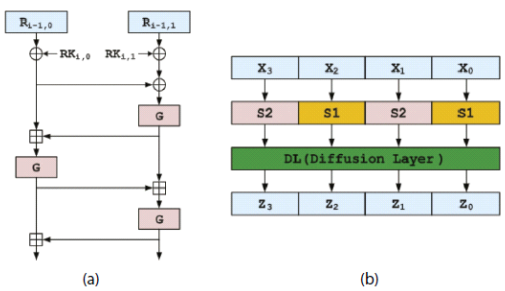


Fig. 1. Structure of SEED (a) F-Function (b) G-Function [18]

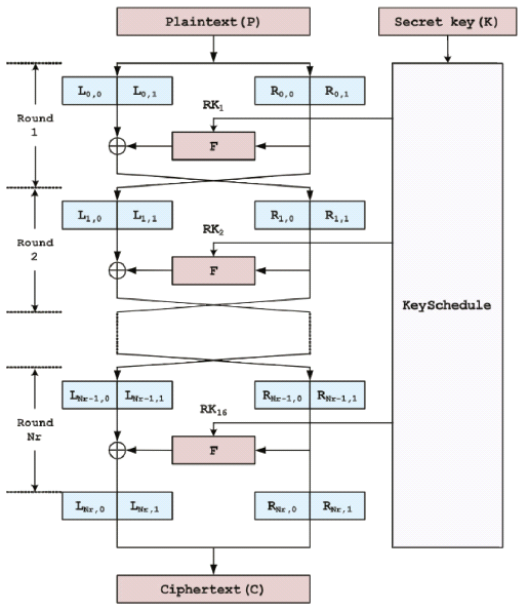


Fig. 2. Structure of SEED [18]

$$A^{(1)} = \begin{bmatrix} 1\,0\,0\,0\,0\,1\,0\,1 \\ 1\,1\,1\,1\,1\,1\,1\,0 \\ 1\,0\,0\,0\,0\,1\,0\,1 \\ 0\,1\,0\,0\,0\,0\,1\,0 \\ 0\,1\,0\,0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,0\,0\,0\,1 \\ 1\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,0\,1\,0\,0 \end{bmatrix}, A^{(2)} = \begin{bmatrix} 0\,1\,0\,0\,0\,1\,0\,1 \\ 1\,0\,0\,0\,0\,1\,0\,1 \\ 1\,1\,1\,1\,1\,1\,1\,0 \\ 0\,0\,1\,0\,0\,0\,0\,1 \\ 1\,0\,0\,0\,1\,0\,1\,0 \\ 0\,0\,0\,1\,0\,1\,0\,0 \\ 0\,0\,0\,1\,0\,1 \end{bmatrix} \qquad (2)$$

2.1.2 LEA

LEA(Lightweight Encryption Algorithm)은 경량 환경에서 기밀성을 제공하기 위해서 개발된 128비트 블록암호 알고리즘이다 [19].

고속으로 암복호화가 가능하도록 ARX(Addition, Rotation, Xor) 기반의 GFN(Generalized Feistel Network) 구조를 가지고 있다. 키 길이는 AES와 동일하게 128/192/256비트 3가지 종류를 사용할 수 있다.

경량 암호 알고리즘으로 특히 ARX 구조로 이루어져있어서 병렬계산에 최적화되어 SIMD 연산을 이용한 CTR 모드를 사용할 시에는 매우 빠르게 암호화 처리가 가능하다. AES-128 CTR과 비교해서 LEA-128 CTR의 경우는 약 2배 가까이 빠른 성능

Table 1. Different key sizes of LEA

Key Size(bits)	Block size(bytes)	Number of Rounds
128	16	24
192	16	28
256	16	32

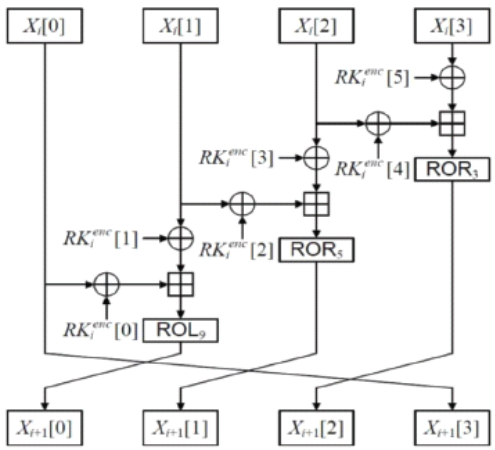


Fig. 3. Round Structure of LEA [19]

을 보인다 [19]. 하지만 이는 연산을 수행하는 메인 CPU에서 ARX 연산을 병렬로 처리할 수 있는 구조가 되어있는 경우에 한정한다. 또한, 32비트의 데이터 Bus를 사용하는 CPU의 경우에는 라운드 함수의 구조상 속도에 최적화 시켜 구현이 가능한다. 키 사이즈에 따라 라운드수는 표 1와 같이 달라진다.

2.1.3 AES

AES(Advanced Encryption Standard)는 DES 암호 알고리즘을 대체하는 알고리즘으로 미국 표준연구소(National Institute of Standards and Technology, NIST)에서 5년간 국제 블록암호 표준의 새로운 암호 알고리즘을 공모를 하여 2001년 11월에 벨기에에서 개발한 Rijndael 암호 알고리즘을 AES로 채택하였다 [20]. 해당 알고리즘은 1977년부터 사용된 DES 암호 알고리즘을 대체하는 용도로 개발이 되었다. AES는 NIST의 FIPS-197 미국 연방정보 처리 표준으로 공표가 된 후에 ISO/IES 18033-3 국제 표준에도 포함되었다. 현재 AES는 국제적으로 또한 군사적으로도 가장 많이 사용되는 블록암호 알고리즘으로 되었다.

Table 2. Different key sizes of AES

Key Size(bits)	Block size(words)	Number of Rounds
128	4	10
192	4	12
256	4	14

AES 이전에 가장 많이 사용되었던 DES와 달리 AES의 구조는 Feistel 구조가 아닌 SPN(Substitution Permutation Network) 구조로 이루어져 있다. SPN에서 Substitution층과 Permutation층을 이용하여 Confusion과 Diffusion을 충족시킨다. 따라서, DES와 달리 AES는 병렬 계산이 가능하지만 복호화와 암호화가 다른 연산을 수행하여 별도로 구현이 필요하다.

2.1.4 ARIA

ARIA는 전자정부 구현을 위해 개발되고 AES와 유사하게 스마트카드, 하드웨어, 임베디드, 소프트웨어 등 다양한 환경에서 사용할 수 있는 암호 알고리즘이다. 국가보안기술연구소 주도로 학계, 국가정보원 등 암호 기술 전문가들이 개발한 국가 블록암호 알고리즘이다 [21]. ARIA는 127/192/256 비트의 비밀키를 사용하고 Involutional SPN 구조를 가진다. 기존의 AES는 암호화와 복호화 별도의 다른 연산을 수행하여 다르게 구현할 필요가 있다. 하지만, Involutional SPN은 암복호화에 다른 연산을 구현할 필요가 없는 구조이다. 따라서, 경량 암호 구현 등에 적합한 구조이다.

ARIA는 S-Box를 이용하여 AES와 동일하게 바이트 단위로 치환을 하고, 확산 단계에서는 16x16 Involution 이진 행렬을 사용한 바이트 간의 확산을 한다. n 라운드 암호화와 복호화 과정은 최초의 라운드 키(eK1)를 적용한 후에 S-Box 대치, 확산, 키 적용 단계를 n-1 라운드 반복한 이후에 최종 단

Table 3. Different key sizes of ARIA

Key Size(bits)	Number of Rounds
128	12
192	14
256	16

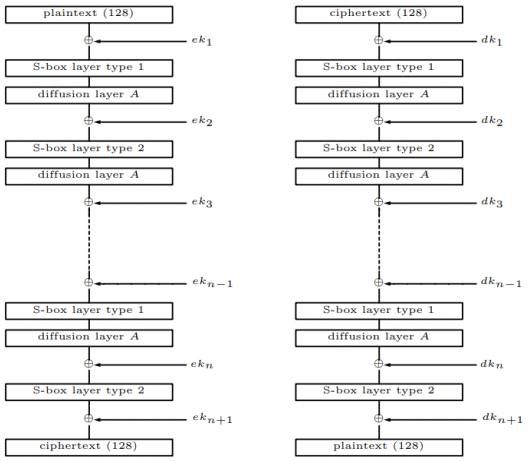


Fig. 4. Structure of ARIA encryption and decryption [21]

계 n라운드에서는 S-Box 치환과 키 적용 단계로만 구성하고 있다.

2.2 구현 방법

각각의 암호 알고리즘에 대해서 ECB 및 CBC 운영모드를 이용하여 구현하였다. ECB 모드는 안전성의 이유로 16바이트 이상의 암호화시에는 실제 제품에서는 사용을 권장하지 않은 운영 모드이다. 하지만, 본 논문에서는 자체 암호 알고리즘의 성능 평가를 위해서 ECB를 모드를 추가하였다. 또한, CBC 모드는 대부분의 암호전용 칩 및 상용 암호모듈에 가장 많이 구현되어 있는 운영모드로 본 논문에서도 16바이트 이상의 암복호화 성능 비교를 ECB 운영모드와 함께 분석을 하기 위해 구현하였다.

또한, 암호 알고리즘 별로 속도 및 메모리 최적화 구현으로 총 6가지 종류의 암호모듈을 구현하였다.

SEED 및 LEA 암호 알고리즘에 대해서는 KISA에서 제공하는 래퍼런스로 구현된 것을 이용하였다.

Table 4. Implementation Methods for Block Cipher

Id.	Algorithm	Optimization
AES_S32	AES	Speed
AES_S8	AES	Memory
ARIA_S32	ARIA	Speed
ARIA_S8	ARIA	Memory
SEED_S32	SEED	Speed
LEA_S32	LEA	Speed

2.2.1 AES_S32

블록암호 알고리즘 AES에 대한 32비트 속도 최적화 구현이다. AES의 표준 정의에서는 8비트 자료형을 기준으로 되어 있으나, 32비트 CPU 에서 처리 속도를 높이기 위해 데이터의 처리를 기본적으로 32비트 자료형으로 수행한다.

특히 복잡한 비트와 바이트 단위의 연산이 필요해지는 SubBytes 계층과 MixColumns 계층을 미리 계산한 32비트값 1024개를 테이블을 이용하여 라운드 함수를 바이트 단위의 테이블 참조 연산만으로 계산한다. 또한 복호화에서는 해당 함수의 역함수를 이용해야 하므로, 역대응 관계값을 1024개 이용하여 계산한다. 따라서 본 알고리즘 구현에서는 연산 로직 이외에 더이상 용량을 줄일 수 없는 8KB의 함수값 테이블을 포함하게 된다.

또한 표준 정의에 비해 복호화 성능을 향상시키기 위해 복호화 라운드키를 사전에 계산하여 이용하는 방법을 이용한다. 이는 실행시간에 점유하는 메모리의 크기를 해당 방법을 이용하지 않은 경우에 비해 약 두 배 크게 한다.

2.2.2 AES_S8

블록암호 알고리즘 AES에 대한 용량 최적화를 포함하는 8비트 구현이다. AES 알고리즘을 AES의 표준정의에서 정의되어 있는 8비트 단위 연산을 이용하여 표준적으로 구현한다. 비록 AES 의 표준에서는 SubBytes 함수를 비트 단위로 연산이 수행이 필요한 비트 계수 다항식 갈루아체 상의 연산대신, 사전에 계산되 8비트 값 256개를 이용하여 계산한다. 또한 그에 대한 역함수도 8비트 값 256개를 이용하여 계산한다. 따라서 본 알고리즘 구현에서는 연산 로직 이외에 더 이상 용량을 줄일 수 없는 512바이트의 함수값 테이블을 포함하게 된다.

본 구현에서는 복호화 연산에 대해서, 실행시간 중에 이용하는 메모리양을 줄이기 위해서 별도의 복호화 라운드키를 사전에 계산하지 않는다. 이는 실행시간에 점유하는 메모리 공간을 줄일 수 있지만, 복호화 연산 처리 속도가 암호화 연산에 비해 최소한 추가 연산만큼 느려지게 된다.

2.2.3 ARIA_S32

블록암호 알고리즘 ARIA 에 대한 32비트 속도 최적화 구현이다. ARIA의 표준 정의에서는 32비트 단위와 하부 8비트 단위로 나뉘는 자료형을 기준으로 처리가 정의되어 있으나, 32비트 CPU 에서 처리속도를 높이기 위해 데이터의 처리를 기본적으로 32비트 자료형으로 수행한다. 특히 S-Box와 확산계층으로 이어지는 암호화 연산에서, 확산계층의 연산을 더 적은 수의 연산으로 처리 하기 위해, S-Box와 몇가지 구성연산을 미리 계산한 32비트값 1024개를 이용한다. 이 값은 복호화 함수에서도 그대로 이용 가능하다.

또한 키 스케줄링 과정에서 암호화 라운드 계산을 수행하기 위해 동일한 구조의 함수테이블로 32비트 값 1024개를 이용한다. 따라서 총합 8KB의 참조테이블을 이용한다. 그러나 테이블화 시킨 연산의 비중이 AES 의 경우에 비해 적기 때문에 암/복호화 연산을 마치기 위해서는 여전히 AES_S32 보다 더 많은 연산을 필요로 하게 된다.

ARIA의 복호화 연산에 이용되는 라운드 키는 시작 라운드와 종료 라운드를 제외하고는 암호화 라운드키에 대해서 확산함수를 한번 적용해야 한다. 이는 구성 연산에서 AES가 일부 가환성을 갖는 것과 달리, ARIA에서는 구성 연산에 대한 가환성이 없기 때문에 발생한다. 따라서 ARIA 알고리즘에서 복호화 라운드키를 별도로 생성하지 않고 암호화 라운드키를 이용하는 것은 라운드수에 비례하는 횟수의 확산함수 연산 만큼의 속도 저하가 발생한다. 따라서 본 구현에서는 복호화 연산에 이용되는 라운드 키를 사전에 생성하여 이용한다. 이는 실행시간에 점유하는 메모리의 크기를 해당 방법을 이용하지 않은 경우에 비해 약 두 배 크게 한다.

2.2.4 ARIA_S8

블록암호 알고리즘 ARIA에 대한 용량 최적화를 포함하는 8비트 구현이다. ARIA는 AES와 마찬가지로 S-Box 가 비트 계수 다항식 갈루아체 상의 연산으로 정의되어 있다. 이는 비트 단위로 많은 연산을 수행해야 하므로, 이를 8비트값 테이블을 이용하여 계산한다. 그러나 ARIA의 S-Box는 2개의 함수와 이에 대한 역함수가 필요하므로, 참조 테이블이 1024개의 값으로 구성된다. ARIA_S32와는 달리 키 스케줄링 과정에서도 동일한 참조 테이블을 이용가능하다. 따라서 본 구현에서는 ARIA_S32에서 포함하는 8KB의 참조 테이블 대신, 순수하게 S-Box 연산을 대체하는 1KB의 참조 테이블을 포함한다.

본 구현에서는 실행시간에 점유하는 메모리의 크기를 줄이기 위해 복호화 연산에 이용되는 라운드키를 사전에 생성하지 않는다. 따라서 복호화가 수행될 때, 암호화 라운드키에 대해 확산함수를 이용하여 복호화 라운드키를 생성하는 과정을 포함한다. 따라서 AES 의 경우에 비해 큰 폭으로 복호화 함수의 성능에 저하가 발생한다.

2.2.5 SEED_S32

블록암호 알고리즘 SEED 에 대한 32비트 속도 최적화 구현이다. SEED 는 비트단위의 선형변환함수를 S-Box 로 하여 G-함수를 구성한다. 따라서 32비트 자료형에 대한 최적화로서 G-함수를 32비트 값 1024개를 이용하는 참조테이블로 구현할 수 있다. 또한 G-함수를 Feistal 구조로 XOR 연산을 통해 적용하므로, 복호화 연산도 G-함수만으로 계산이 가능하다. 본 구현에서는 4KB의 참조 테이블을 이용하여 G-함수에 대한 최적화를 하였다. 이러한 G-함수를 이용한 Feistal 구조 덕분에 SEED는 암호화 라운드키와 복호화 라운드키가 동일한 구조를 갖는다. 또한 16라운드 동안 각각 8바이트씩만 이용하므로 다른 블록암호 알고리즘에 비해 적은 메모리를 실행 시간동안 점유한다.

2.2.6 LEA_S32

블록암호 알고리즘 LEA 에 대한 32비트 속도 최적화 구현이다. LEA 알고리즘은 다른 암호 알고리즘과 달리 암호화 연산이 모두 32비트 단위의 자료형을 그대로 이용할 수 있는 연산에만 의존한다. 따라서 암호화 연산에 별도의 테이블을 이용한 참조 최적화를 수행하지 않아도 다른 암호 알고리즘보다 빠른 처리가 가능하다. 오히려 최신의 CPU를 위한 128비트 레지스터나 256비트 레지스터를 이용한 SIMD(Single Instruction Mutiple Data) 연산에 기반한 최적화도 존재한다. 그러나 Pixhawk4는 그러한 연산 최적화가 불가능한 환경이다.

단순한 라운드 함수 연산 대신 LEA 알고리즘은 라운드당 라운드키를 24바이트 사용하며, 라운드 수

도 128비트 키에 대해서 24개로 다른 알고리즘에 비해 라운드 수가 많다. 따라서 라운드키를 사전에 생성해서 이용하는 경우 다른 블록암호 알고리즘에 비해 많은 메모리를 실행 시간동안 점유하게 된다. 라운드키 생성 방법은 단순하기 때문에, 라운드키를 사전에 생성하지 않고 실행시간에 계산해서 이용하는 방법도 존재한다. 본 구현에서는 속도 최적화를 위해 사전에 생성된 라운드키를 이용한다.

2.3 하드웨어 사양

본 연구에서 무인 항공기의 제어 오픈소스 프로젝트에서 가장 많이 사용되고 있는 PX4 Platform을 이용하였다. 구현 시 이용한 commit 버전은 tags/v1.12.0을 사용하였다 [17]. PX4가 호환되는 메인 비행 컨트롤러의 하드웨어는 FMUv5를 사용하였다(Fig. 5). FMUv5는 Holybro와 PX4팀이 협력하여 설계 및 제작한 고급 자동 조종장치로 PX4 소프트웨어 v1.7이상의 버전 실행에 최적화 되어있고 FMUv5는 NuttX의 OS에서 PX4가 동작한다. 구체적인 하드웨어 사양은 이하와 같다.

- 메인 FMU 프로세서 : STM32F765(32-Bits ARM Coretex-M7 with 217MHz)
- FLASH :2MB
- SRAM : 512KB
- IO Processor : STM32F100(32-Bits ARM Coretex-M3 with 32MHz)

Fig. 5. FMUv5

III. 결 과

3.1 프로그램 용량

프로그램 용량은 암호 알고리즘에 대한 소스 코드가 컴파일되어 펌웨어에 포함되는 바이너리 데이터의 크기로 암호 알고리즘이 포함되기 전과 후의 두가지 펌웨어의 차이로 계산하였다.

프로그램 용량에 대해서 LEA_S32를 제외한 다른 알고리즘은 참조 테이블을 이용하거나, 별도의 복잡한 연산 함수를 정의하게 되므로 큰 차이를 보인다. 기본적으로 32비트 테이블을 이용하는 경우 실행 코드보다 참조 테이블 크기의 비중이 커지고, 8비트 테이블을 이용하는 경우에는 참조 테이블 크기보다, 복잡한 연산 정의를 위한 실행 코드의 비중이 커진다. 따라서 참조 테이블 크기의 차이에 비해 실제 프로그램 크기의 차이는 크지 않은 편이었다. 암호모듈은 전체 Flash(2MB)의 약 0.5 ~ 0.8%를 차지하였다. CBC 운영모드의 경우에는 IV 연산을 위한 함수 등이 추가됨으로써 ECB 운영모드에 비해서 약 1.1%가량 증가하였다.

3.2 실행 메모리 용량

실행 메모리 용량은 암호 알고리즘을 실행하기 위해 힙 공간에 할당이 필요한 메모리 공간의 크기로 라운드 키와 같이 암복호화 동작에 필요한 매개 변수를 보관하고 있다.

실행시에 고정적으로 점유하게 되는 메모리크기는 암호알고리즘에 크게 의존한다. 또한 암호화 라운드키와 복호화 라운드키가 서로 다른 경우, 라운드키의 생성 정책에 의해 크게 좌우되는 결과를 확인할 수 있었다. CBC의 경우는 IV값 및 관련 메모리가 추가적으로 필요하여 ECB 운영모드에 비해서 약 10.1%정도 증가하였다. 하지만 전체적으로 전체 메모리 용량(512KB)의 0.02 ~ 0.1%를 사용하였다.

3.3 암복호화 속도

암복호화 속도는 암호화 및 복호화시에 속도를 측정한 것으로 1회당 64바이트(2블록)의 데이터를 10000회 암호화 및 복호화를 하여 처리속도 S는 다음과 같이 측정하였다.

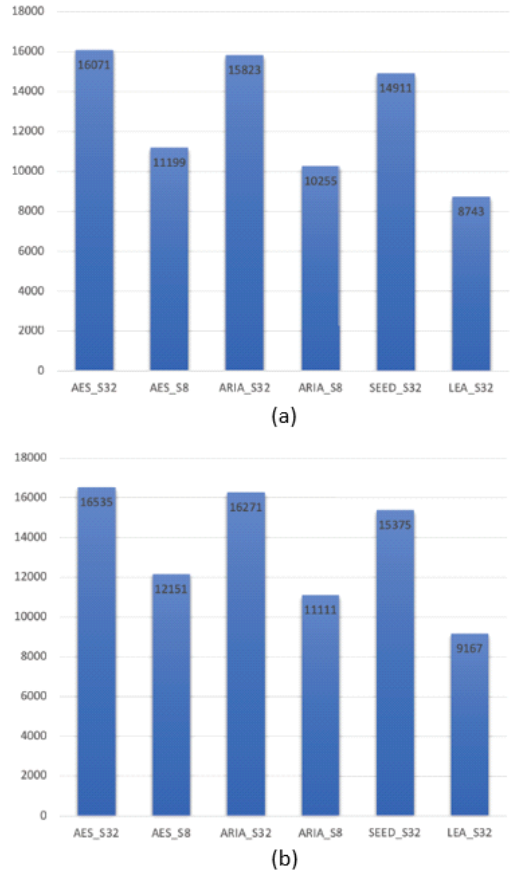


Fig. 6. Results of program memory size (bytes) (a) ECB Mode, and (b) CBC Mode

$$S(MB/s) = \frac{\text{전체 처리 데이터 크기}(MB)}{\text{소요시간}(s)} = \frac{\text{1회 처리 데이터 크기} \times \text{반복횟수}}{\text{소요시간}(s)}$$

실제로 암호모듈이 활용되는 데이터는 비디오 영상의 암복호화에 적용이 예상된다. 따라서 해당 비디오 영상 데이터의 비트레이트와 단위를 맞춰서 암호모듈로 영상 데이터의 실시간 암복호화가 가능한지 비교하기 위해서 MB/s 단위를 사용하였다.

암호화 속도는 32비트 연산이나 참조 테이블의 비중이 높을수록 빠른 경향을 보인다. 특히 참조 테이블을 이용하지 않고 32비트 자료형의 연산만으로 구현 가능한 LEA_S32가 가장 빠른 결과를 보였다. 8비트 참조 테이블만을 이용하는 경우에는 상대적으로 프로그램 용량에서 얻는 이득보다는, 암호화 속도의

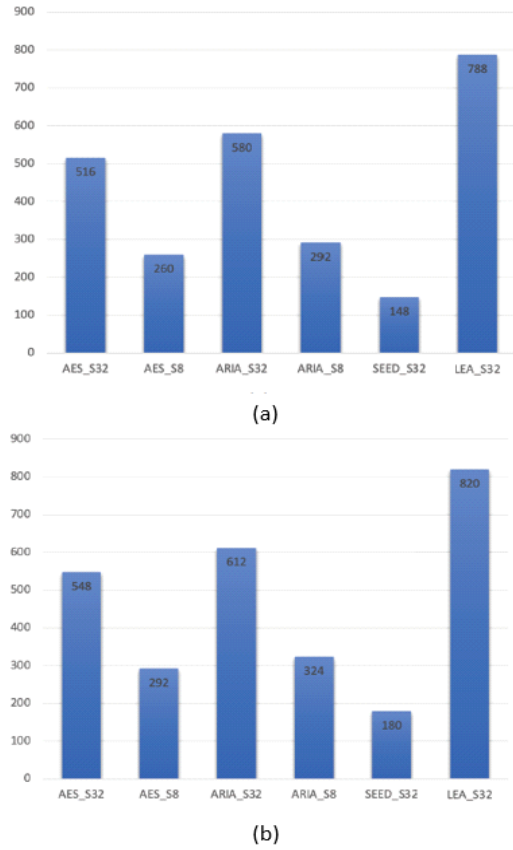


Fig. 7. Results of RAM Size (bytes) (a) ECB Mode, and (b) CBC Mode

저하가 압도적으로 영향이 큰 경향을 보였다.

블록암호의 복호화 함수는 암호화 함수의 역함수로, 대부분의 블록암호 알고리즘에서 서로 비슷한 구조를 갖게 된다. 따라서 속도도 거의 비슷한 경향을 갖게 된다. 그러나 복호화 라운드키의 생성에 연산이 더 많이 필요하게 되는 ARIA 알고리즘은 사전에 복호화 라운드키를 생성하지 않는 경우에 암호화 함수와 복호화 함수의 성능차이가 극명하게 발생했다.

전체적으로 각 암호 모듈별로 암복호화 속도의 비율을 유사하였다. CBC 운영 모드의 경우는 IV 관련 캐시 및 XOR 관련 연산 등의 추가적인 연산이 포함되어 있어, CBC 운영 모드의 경우가 ECB 운영 모드에 비해서 약 3.3% 정도 속도가 감소되었다. 특히 다른 암호 알고리즘에 비해서 처음부터 고속화 암호 알고리즘으로 구조적으로 구현된 LEA의 경우가 약 1.8배에서 9.5배 정도 차이가 발생했다.

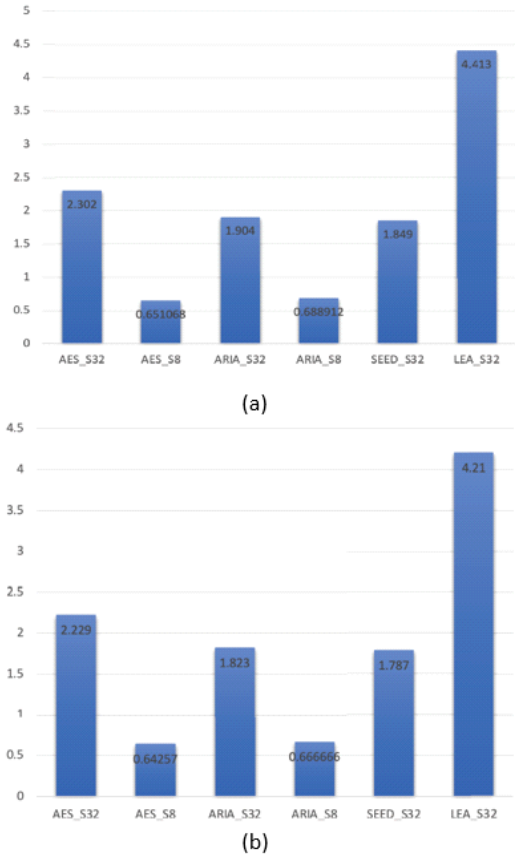


Fig. 8. Encryption Speed (MBytes/s) (a) ECB Mode, and (b) CBC Mode

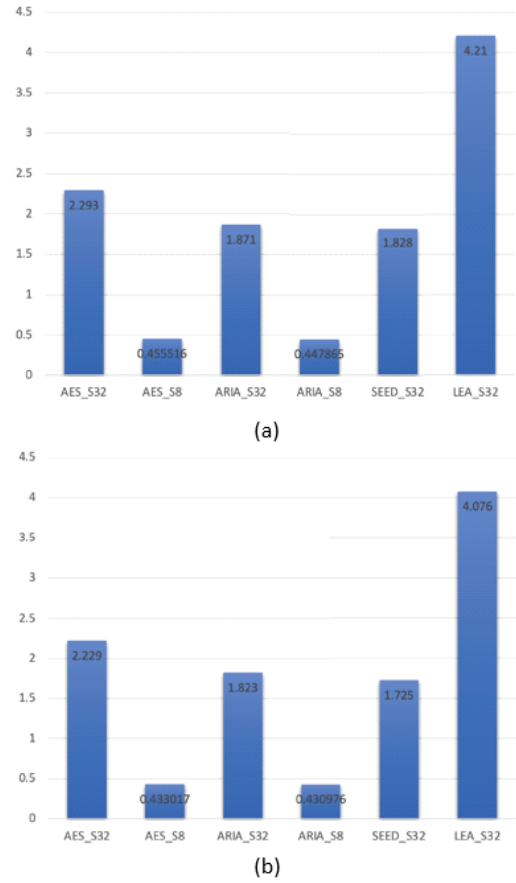


Fig. 9. Decryption Speed (MBytes/s) (a) ECB Mode, and (b) CBC Mode

3.4 고찰

본 연구에서는 4종의 블록암호 알고리즘을 소프트웨어로 구현한 암호모듈을 무인 항공기의 플랫폼 PX4에 적용하여 성능 평가를 수행하였다. 그 결과 프로그램 메모리는 LEA의 암호모듈이 약 8KB로 가장 작은 용량을 차지하였다. 이는 LEA의 암호 알고리즘은 타 블록암호와 다르게 S-Box 등이 존재하지 않고 구조적으로 단순한 ARX(Addition, Rotation, Xor)만을 이용하기 이러한 결과가 나왔을 것으로 추정한다. 실행 메모리적인 측면에서는 SEED 암호모듈이 가장 적은 실행 메모리를 사용하였다. 특히 가장 많이 사용한 LEA 비해서는 약 22% 정도만 필요하였다. 이는 본 연구에서 구현된 것은 라운드 키가 LEA는 사전 정의된 방식이지만 SEED의 경우는 On-the-fly 방식으로 각 라운드별로 생성하고 구조적으로 실행 메모리를 덜 쓰는 Feistel 구조를 사용하였기 때문으로 분석된다. 암복호화 속도면에서는 처음부터 속도 위주로 만들어진 암호 알고리즘인 LEA가 타 블록암호모듈에 비해서 최대 8배 이상의 차이를 보였다.

LEA 암호모듈의 암호화 속도는 약 4.776MB/s로 16바이트 암호화시에 약 3.35us정도로 가능하다. 이를 통신 속도로 변경하면 약 38.208Mbps로 만약에 통신 구간 상에서의 속도가 38Mbps보다 느릴 경우에는 통신상에서의 병목현상이 발생한다. 드론에서 사용되는 Wifi 네트워크에서의 통신 속도, 거리에 관한 실험 논문을 참고하면 경우에 따라서 암복호화 속도보다 더 느린 네트워크 속도로 인하여 암복호화 이외에 통신에서의 병목 현상이 발생할 가능성이 있음을 보였다 [22].

또한, 실제 드론의 카메라를 통해서 촬영한 38초 분량의 30FPS의 Full HD 영상의 비트레이트를

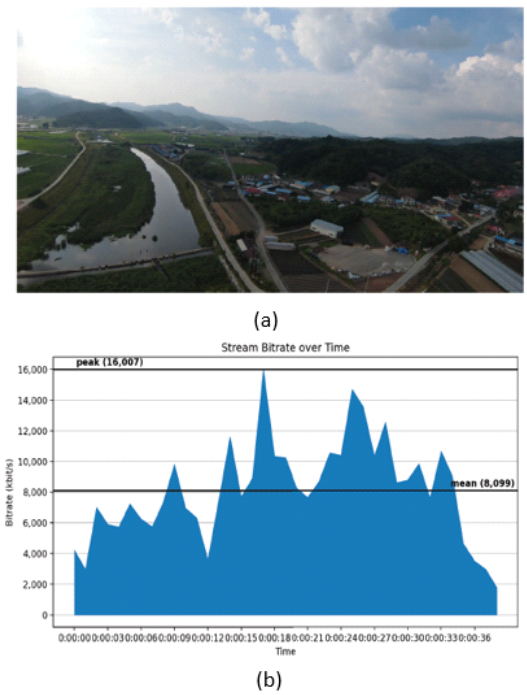


Fig. 10. Sample Video from UAV (a) Sample Captured Image (b) Stream Bitrate over Time

분석하면 Fig. 10. (b)와 같이 대체적으로 움직임이 적는 곳은 4Mbps, 움직임이 많은 곳은 최대 약 16Mbps를 나타냈었다. 평균적으로 약 8Mbps의 비트레이트로 LEA의 암복호화 속도보다 낮아 충분히 실시간 암복호화가 가능할 것으로 생각된다. 만약에 4K 등의 영상의 경우에는 38Mbps보다 높은 비트레이트를 필요하여 암호화 혹은 복호화로 인하여 영상의 지연이 발생할 가능성이 있다. 하지만, 4K의 영상을 실시간으로 통신 네트워크를 이용하여 지상으로 전달하는 것은 대부분의 드론에서 높은 데이터 전송이 필요하여 수행되지 않고 드론 내부의 별도의 저장 장치(예, SD카드)에 별도로 저장되어지고 지상으로 전달되는 영상의 경우는 SD 혹은 HD 화질의 데이터가 전송되는 것이 일반적이다.

IV. 보안 UAV의 평가 기준

현재 UAV에서 주로 사용되는 PX4 오픈 플랫폼에서의 보안 기능은 제공되지 않고 있다. 따라서 본 논문에서는 통신 채널의 데이터의 기밀성을 위해서 다양한 종류의 암호모듈을 구현한 것에 대한 성능 평가를 하였다. 이러한 암호모듈을 이용한 보안성을 제공하는 보안 UAV에서의 평가 기준에 대해서 논하고자 한다. 현재 보안 IT 제품에서 활용되는 것에서 가장 많이 사용되는 것이 국제공통평가기준(Common Criteria, CC)에 따른 보안기능요구사항[23] 및 암호모듈검증제도이다. 두 가지 제도 모두 국제 기준으로 국내에서도 이를 적용한 국내용 CC 및 KCMVP라는 이름으로 국내 보안 IT 제품에 대한 평가를 위해서 활용되고 있다.

4.1 보안기능요구사항

보안기능요구사항(Security Functional Requirements, SFR)은 보안 UAV라는 평가 대상(Target Of Evaluation ,TOE)의 보안 목적(Security Objective) 표준화된 언어로 표현한 것이다. 보안기능요구사항에 앞서 본 연구 대상의 보안 UAV에서의 보안 목적은 다음과 같다.

- 저장 데이터 보호 : TOE에 저장된 영상 및 센서 등의 데이터는 인가되지 않은 노출, 변경, 삭제로부터 보호해야 한다.
- 데이터 전송 보호 : TOE는 구성 요소간에 전송되는 데이터에 대해서 기밀성 및 무결성을 보장해야한다.
- 안전한 암호 : TOE는 데이터를 암호화하기 위해서는 암호키를 안전하게 생성 및 관리해야한다. 또한, 이때에 사용되는 암호 알고리즘은 안전성이 보장된 암호 알고리즘을 이용해야한다.
- 네트워크 정보 흐름 통제 : 무선 통신에서 비정상적인 패킷을 차단하여 불법적인 접근을 통제해야한다.
- 시간 동기화 : 신뢰성있는 시간 정보를 TOE가 제공해야한다.
- 안전한 업데이트 : TOE는 소프트웨어 무결성을 보장하여 업데이트가 처리될 수 있도록 해야한다.
- 식별 및 인증 : TOE에 대해서 접근하는 사용자에 대해서 식별해야 하고 식별된 사용자에 따라서 다른 접근 권한을 인증 통해서 제공되어야 한다.
- 감사 및 관리 : TOE의 인가된 관리자가 TOE를 효율적으로 관리할 수 있는 관리 수단을 제공해야한다. 또한, TOE는 보안과 관련

된 사건을 기록 및 유지해야하며 기록된 데이터를 확인할 수 있는 수단을 제공해야한다.

- 재사용 공격 대응 : TOE는 제어 신호 등의 재사용 공격에 대응하기 위한 방안이 마련되어야 한다.

상기의 보안 목적에 따라서 도출된 보안기능요구사항은 다음과 같다.

4.1.1 보안 감사

보안 감사(Functional Security Audit, FAU)는 보안 관련 행동에 관련된 정보의 인식, 기록, 저장, 분석을 포함한다.

- FAU_ARP.1(보안 경보)
- FAU_GEN.1(감사 데이터 생성)
- FAU_SSA.1(보안 감사분석)
- FAU_STG.1(감사 증적 저장소 보호)
- FAU_STG.3(감사 데이터 손실 예측 시 대응행동)

4.1.2 암호 지원

암호 지원(Functional Cryptographic Support, FCS)는 TOE가 암호기능을 구현할 경우 사용되며 하드웨어, 펌웨어, 소프트웨어로 구현될 수 있다.

- FCS_CKM.1(암호키 생성)
- FCS_CKM.2(암호키 분배)
- FCS_CKM.4(암호키 파기)
- FCS_COP.1(암호 연산)

4.1.3 사용자 데이터 보호

사용자 데이터 보호(Functional User Data Protection, FDP)는 사용자 데이터 보호와 관련된 사항을 포함한다.

- FDP_SDI.1(저장된 데이터의 무결성)
- FDP_ITT.1(TOE 내부전송)
- FDP_UCT.1(TSF간 전송되는 사용자 데이터 비밀성)
- FDP_UIT.1(TSF간 전송되는 사용자 데이터 무결성)
- FDP_IFF.1(정보흐름 통제)
- FDP_IFC.1(정보흐름통제 정책)

4.1.4 식별 및 인증

식별 및 인증(Functional Identification & Authentication, FIA)는 사용자의 신원을 설정하고 증명하기 위한 사항을 포함한다.

- FIA_AFL.1(인증 실패)
- FIA_SOS.1(비밀정보의 검증)
- FIA_SOS.2(비밀정보의 생성)
- FIA_UAU.1(사용자 인증)
- FIA_UID.1(사용자 식별)

4.1.5 TSF 보호

TSF 보호(Functional Protection of the TSF, FPT)는 TSF를 구성하는 메커니즘의 무결성과 관리에 관련된 사항을 포함한다.

- FPT_RPL.1(재사용 공격 탐지 및 대응행동)
- FPT_STM.1(신뢰할 수 있는 타임스탬프)
- FPT_TDC.1(TSF 간 전송되는 TSF 데이터의 기본적인 일관성)
- FPT_TST.1(TSF 자체 시험)
- FPT_ITT.1(TSF 데이터 내부전송)
- FPT_ITA.1(외부전송 TSF 데이터의 가용성)
- FPT_ITC.1(외부전송 TSF 데이터의 비밀성)
- FPT_ITI.1(외부전송 TSF 데이터의 무결성)

4.1.6 보안관리 클래스

FMT(Functional Security Management, FMT)는 보안속성, TSF 데이터, TSF 기능 등 TSF의 여러 관련 사항을 관리하는 것을 포함한다.

- FMT_MSA.1(보안속성 관리)
- FMT_MSA.3(정적 속성 초기화)
- FMT_SMF.1(관리기능 명세)
- FMT_SMR.1(보안 역할)
- FMT_MTD.1(TSF 데이터 관리)

V. 결 론

본 연구에서는 UAV에서 기밀성을 제공하기 위해서 AES, ARIA, SEED, LEA의 블록암호를 구현한 암호모듈을 PX4 플랫폼에 탑재하였다. 또한 각

각의 암호모듈에 대한 메모리, 속도 등으로 성능 분석을 하여 UAV에 보안을 적용하기 위해서 적합한 암호모듈에 대해서 논의하였다. 그 결과 만약에 UAV가 높은 속도의 암복호화가 필요할 시에는 LEA를 이용하는 것이 가장 좋은 결과를 보였다.

또한, UAV의 제한된 RAM 용량에 따라서 최소한의 리소스를 위한 암호모듈로는 8비트로 구현한 AES 및 SEED 암호모듈을 사용하는 것이 LEA에 비해서 약 20% 이내의 리소스를 절약할 수 있는 결과를 나타내었다. 이러한 결과는 UAV 어플리케이션의 환경에 따라서 어떠한 암호모듈 구현법 및 알고리즘을 사용해야하는 지에 대한 판단의 근거가 된다.

또한, 마지막으로 UAV에서의 다양한 보안 위협에 대응하기 위해서 국제공통평가기준을 통한 보안요구사항을 도출하였다. 특정한 제품군에 대한 보안요구사항의 집합인 보호프로파일 또한 UAV에 대해서는 마련되어 있지 않은 상황이다.

본 연구에서는 기밀성을 위해서 블록암호 알고리즘만 구현한 암호모듈에 대한 성능 평가를 수행하였다. 하지만 기밀성 이외에 무결성, 부인방지, 인증 등을 위해서는 공개키 암호 및 키 교환, 해시 함수 등 다양한 암호 알고리즘을 구현해서 UAV의 플랫폼인 PX4에 통합하여 구현되어야할 필요성이 있다.

References

[1] Koubâa, A., Allouch, A., Alajlan, M., Javed, Y., Belghith, A., and Khalgui, M., "Micro air vehicle link (mavlink) in a nutshell" A survey. IEEE Access 7, pp. 87658 - 87680, 2019

[2] Khan, N.A., Jhanjhi, N.Z., Brohi, S.N., and Nayyar, A., "Emerging use of uav's: secure communication protocol issues and challenges" In: Drones in smart-cities, pp. 37 - 55, 2020

[3] Hassija, V., Chamola, V., Agrawal, A., Goyal, A., Luong, N.C., Niyato, D., Yu, F.R., Guizani, and M., "Fast, reliable, and secure drone communication" A comprehensive survey. IEEE Communications Surveys & Tutorials, 2021

[4] Shoufan, A., AlNoon, H., and Baek, J., "Secure communication in civil drones" In: International Conference on Information Systems Security and Privacy. pp. 177 - 195, 2015

[5] Rani, C., Modares, H., Sriram, R., Mikulski, D., Lewis, F.L., "Security of unmanned aerial vehicle systems against cyber-physical attacks" The Journal of Defense Modeling and Simulation 13(3), 331 - 342 (2016)

[6] Marty, J.A., "Vulnerability analysis of the mavlink protocol for command and control of unmanned aircraft" Tech. rep. (2013)

[7] Hartmann, K., and Steup, C., "The vulnerability of uavs to cyber attacks-an approach to the risk assessment" In: 2013 5th international conference on cyber conflict (CYCON 2013). pp. 1 - 23, 2013

[8] Iqbal, S., "A study on uav operating system security and future research challenges" In: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC). pp. 0759 - 0765, 2021

[9] Son, Y., Shin, H., Kim, D., Park, Y., Noh, J., Choi, K., Choi, J., and Kim, Y., "Rocking drones with intentional sound noise on gyroscopic sensors" In: 24th USENIX Security Symposium (USENIX Security 15). pp. 881 - 896, 2015

[10] Arteaga, S.P., Hernández, L.A.M., Pérez, G.S., Orozco, A.L.S., and Villalba, L.J.G., "Analysis of the gps spoofing vulnerability in the drone 3dr solo" IEEE Access 7, 51782 - 51789, 2019

[11] Dey, V., Pudi, V., Chattopadhyay, A., and Elovici, Y., "Security vulnerabilities of unmanned aerial

vehicles and countermeasures: An experimental study" In: 2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID). pp. 398 - 403, 2018

[12] Kwon, Y.M., Yu, J., Cho, B.M., Eun, Y., and Park, K.J. "Empirical analysis of mavlink protocol vulnerability for attacking unmanned aerial vehicles" IEEE Access 6, 43203 - 43212, 2018

[13] Domin, K., Symeonidis, I., and Marin, E., "Security analysis of the drone communication protocol" Fuzzing the mavlink protocol, 2016

[14] Allouch, A., Cheikhrouhou, O., Koubâa, A., Khalgui, M., and Abbes, T., "Mavsec: Securing the mavlink protocol for ardupilot/px4 unmanned aerial systems" In: 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). pp. 621 - 628, 2019

[15] Jeong, S., Park, E., Seo, K.U., Do Yoo, J., and Kim, H.K., "Muvids: False mavlink injection attack detection in communication for unmanned vehicles" In: Workshop on Automotive and Autonomous Vehicle Security (AutoSec). vol. 2021, pp. 25, 2021

[16] Atoev, S., Kwon, O.J., Kim, C.Y., Lee, S.H., Choi, Y.R., and Kwon, K.R. "The secure uav communication link based on otp encryption technique" In: 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN). pp. 1 - 3, 2019

[17] PX4 github. Website. [Online] https://github.com/PX4/PX4-Autopilot.git, accessed: 2022-03

[18] TTAS.KO-12.0004/R1, "128-bit Block Cipher SEED", 2005

[19] D. Hong, J.-K. Lee, D.-C. Kim, D. Kwon, K. H. Ryu, and D.-G. Lee, "LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors," Lect Notes Comput Sc, pp. 3 - 27, 2014

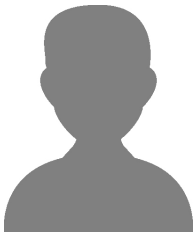
[20] Dworkin, M. J., Barker, B., Nechvatal, J. R., Foti, J., Bassham, L. E., Roback, E., and Dray Jr, J. (2001). "Advanced Encryption Standard (AES)", NIST FIPS-197.

[21] Kwon, D., Kim, J., Park, S., Sung, S. H., Sohn, Y., Song, J. H., ... and Hong, J. "New block cipher: ARIA". In International conference on information security and cryptology pp. 432-445, 2003

[22] Guillen-Perez, A., Sanchez-Iborra, R., Cano, M.D., Sanchez-Aarnoutse, and J.C.,Garcia-Haro, J., "Wifi networks on drones" In: 2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT). pp. 1 - 8, 2016

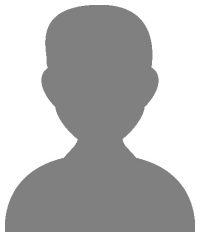
[23] Common Criteria for Information Technology Security Evaluation, version 3.1, revision 5. part 2: Functional Security Components. 2017

〈저자소개〉

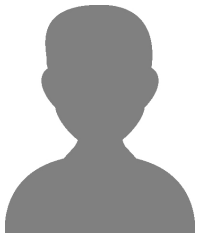


김 용 대 (Yongdae Kim) 정회원
2004년 2월: 도호쿠 대학교 전자전기공학부 학사
2010년 2월: 도호쿠 대학교 정보과학연구과 석사
2010년 3월~2011년 6월: 소니(SONY) 연구원
2011년 7월~현재: ETRI부설연구소 선임연구원
〈관심분야〉 사이버보안, 정보보호, 인공지능

김 덕 진 (Deokjin Kim) 정회원
2004년 2월: 인하대학교 컴퓨터공학과 학사
2006년 2월: 포항공과대학교 컴퓨터공학과 석사
2019년 8월: 한국과학기술원 전산학부 정보보호대학원 박사
2006년 1월~2007년 9월: LG전자 연구원
2007년 9월~현재: ETRI부설연구소 책임연구원
〈관심분야〉 시스템 보안, 네트워크 보안, 정보보호



이 은 경 (Eunkyoung Yi) 정회원
2012년 9월~현재: ETRI부설연구소 책임연구원
〈관심분야〉 시스템 보안, 네트워크 보안, 정보보호



이 상 욱 (Sangwook Lee) 정회원
1999년 2월: 경북대학교 컴퓨터과학과 학사
2001년 8월: 경북대학교 컴퓨터과학과 석사
2004년 2월: 경북대학교 컴퓨터과학과 박사수료
2004년 2월~현재: ETRI부설연구소 책임연구원/실장
〈관심분야〉 사이버보안, 취약점분석·평가, 드론·UAM 보안