논문 21-46-11-14 The Journal of Korean Institute of Communications and Information Sciences '21-11 Vol.46 No.11
https://doi.org/10.7840/kics.2021.46.11.1882

위성항법 민간신호의 인증

조 태 남•°, 용 승 림*, 정 원 찬**, 이 상 욱**, 유 준 규**

Authentication Scheme for Civil Signals of Navigation Satellites

Taenam Cho•°, Seunglim Yong*, Wonchan Jung**, Sanguk Lee**, Ryu Joon Gyu**

요 약

우리는 미국의 범지구위성항법시스템인 GPS (Global Positioning System)으로부터 위치 및 시간 정보를 수신하여 네비게이션 등에서 사용하고 있다. 현재 GPS를 포함한 GNSS에 대한 다양한 공격이 꾸준히 발생하고 있으며 점차 증가할 것으로 예상되고 있다. 유럽의 Galileo와 미국의 GPS 등은 민간신호의 인증 서비스를 추진하고 있으며 다른 위성항법 시스템에 대해서도 연구가 이루어지고 있다. 따라서 우리나라에서 추진하고 있는 독자적인 차세대 위성항법시스템인 KPS (Korea Positioning System)에 대해서도 민간 신호의 보호 방안이 요구된다. 본 논문에서는 KPS에 적용될 수 있는 민간용 위성항법 데이터의 인증 방안을 제시한다.

키워드 : 위성항법시스템, KPS, 보안, 메시지 인증
Key Words : GNSS, KPS, Satellite Security, Message Authentication

ABSTRACT

We receive location and time information from the Global Positioning System (GPS), a global navigation satellite system in the United States, and use it in services such as navigation. Currently, attacks on GNSS, including GPS, are occurring steadily and are expected to gradually increase. Galileo in Europe and GPS in the US are promoting civil signal authentication services, and other satellites are also being studied. Therefore, a method for protecting civilian signals is also required for the Korea Positioning System (KPS), which is an independent next-generation navigation satellite system that is being promoted in Korea. In this paper, we propose an authentication method for navigation data for civil use that can be applied to KPS.

Ⅰ. 서 론

우리가 사용하고 있는 GPS (Global Positioning System)와 같은 위성항법시스템은 인공위성이 전송하는 항법메시지(Navigation Message)를 이용하여 단말기가 현재 시각과 자신의 위치를 결정할 수 있도록 해주는 체계로서 실생활과 밀접한 다양한 분야에서 활용되고 있다. GNSS 디바이스 및 서비스 시장은 2020년에 1,629억 달러로 평가되었으며 2021-2026년 기간 동안 14.6%의 CAGR을 등록하여 2026년까지 3,531억 7,000만 달러에 이를 것으로 예상된다[1]. GNSS를 이용한 서비스는 더욱 다양해질 것이며, 초연결성을 가지는 4차 산업혁명에 힘입어 육상교통, 정보통신, 우주항공, 해양, 농업, 문화, 레포츠 등 다양한 분야에

※ 본 연구는 2021년도 한국전자통신연구원 연구운영비지원사업의 재원으로 수행하고 있는 한국전자통신연구원의 연결의 한계를 극복하는 초연결 입체통신 기술 연구 (21ZH1100)의 지원으로 수행한 정밀 보강 위성항법 서비스 사용자 인증방법 연구 과제의 결과입니다.
•° First and Corresponding Author : Woouk University Department of IT & Electronics Engineering, tncho@woosuk.ac.kr, 정회원
* Inha Tech. College, slyong@inhatc.ac.kr, 정회원
** Electronics & Telecommunications Research Institute, wcjung@etri.re.kr, 정회원; slee@etri.re.kr, 정회원; jgryurt@etri.re.kr
논문번호 : 202108-196-B-RE.R1, Received August 3, 2021; Revised August 26, 2021; Accepted August 27, 2021

서 고정밀의 위성항법시스템의 활용도가 더욱 높아질 것이다[2]. 우리나라에서도 4차 산업분야의 다양한 분야에서 위성항법시스템이 활용될 것으로 예상되고 있다[3].

한편, 전세계적으로 이러한 위성항법시스템에 대한 다양한 공격이 현실화 되고 있으며[4-6], 우리나라에서도 GPS 스푸핑(Spoofing) 공격이 보고된 바 있다[7]. 위성항법시스템을 이용한 서비스 영역별로 인증, 정확도, 무결성, 가용성 등의 중요성이 다르지만 가장 활용도가 높은 LBS (Location Based Service) 영역에서 인증이 가장 중요한 요소 중의 하나로 평가된다. 예로, 일반 road navigation은 인증이 덜 중요한 편이나 자율주행 자동차 등에서는 위치 정보의 정확도뿐만 아니라 신호의 인증이 매우 중요한 요소이다. 이 외에도 범인 추적, 위치기반 과금, 긴급 구조 호출, 밀입국 제어, 헬스 케어 등에서 소소한 생활의 불편을 넘어 개인과 사회의 안전을 위협할 뿐만 아니라 국가적 범죄에 영향을 미칠 수 있다. 그럼에도 불구하고 군용 신호에서만 인증과 암호화 기능이 지원되고 있어, 민간 사용자 보호를 위한 다양한 위성 신호 인증 방식이 연구되고 있다. 갈릴레오 위성은 2030년 2세대 Galileo를 통해 민간 항법메시지 인증을 지원하는 것을 목표로 테스트에 성공하였으며[8] GPS도 인증 서비스를 준비하고 있다[9].

항법메시지 인증은 메시지 레벨과 신호레벨에서 이루어질 수 있다. 본 연구는 KPS에 적용할 수 있는 메시지 레벨에서의 스푸핑 공격을 막기 위한 인증 기법을 제안한다. 2장에서는 인증에 사용되는 암호기술의 개념에 대해 기술하고 3장에서는 위성항법시스템과 인증 현황에 대해 기술한다. 4장에서는 본 연구에서 제안하는 인증 기법을 제안하고 5장에서는 제안 기법에 대한 분석과 비교를 하며 6장에서 결론을 맺는다.

Ⅱ. 기반 암호 기술

2.1 암호시스템(cryptosystem)

공공 네트워크에서 메시지를 보호하기 위한 기본적인 기술은 암호기술이다. 안전한 통신을 위해서는 통신 참가자들이 난수(random number)인 키(key)를 소유하게 된다. 비밀키 암호방식은 송수신자만이 하나의 비밀키를 공유한다. AES(Advanced Encryption Standard)[10], SEED[11] 등이 대표적이다. 공개키 암호방식에서는 각 참가자가 각각 자신의 공개키 PK와 개인키 RK 한쌍을 소유하는데 PK는 공개값이고 RK는 자신만 소유하는 비밀값이다. RSA(Rivest-Shamir-Adleman)[12], ECC(Elliptic Curve Cryptography)[13] 등이 대표적이다. 메시지를 암호화하거나 인증하기 위해서 메시지와 키를 암호 알고리즘에 적용시킨다.

2.2 해시함수(hash function)

해시함수는 메시지 m에 대하여 m의 길이와 상관없이 일정한 짧은 비트 길이의 값을 산출한다. 특히 암호학적 해시함수는 일방향성(one-wayness), 충돌회피성(collision resistent)과 같은 몇 가지 엄격한 조건을 만족하는 해시함수이다. 즉, 해시값으로부터 원문 m을 알아낼 수 없고, 서로 다른 해시값은 매우 높은 확률로 다른 메시지에 대한 해시값이라고 결론지을 수 있다. 이러한 성질 때문에 해시값을 메시지에 대한 다이제스트(digest) 혹은 지문(fingerprint)이라고 부른다. 대표적인 해시함수는 SHA1(Secure Hash Algorithm 1)[14], SHA2(Secure Hash Algorithm 2)[15] 등이 있으며, 산출되는 해시값의 비트 길이가 길수록 안전하다.

해시함수는 공개된 함수로서 누구나 m에 대한 해시값을 구할 수 있다. 여기에 비밀키 k를 입력 파라미터로 사용하는 함수가 키해시(keyed hash function)함수이다. 키해시로 생성된 값은 이 k를 공유한 개체만이 검증할 수 있다. 대표적인 함수로는 HMAC (Hash-based Message Authentication Code)[16], CMAC (Cypher-based Message Authentication Code)[17] 등이 있다.

2.3 메시지 인증(Message Authentication)

위성항법메시지 인증에 사용되는 대표적인 암호 기술로서 TESLA(Timed Efficient Stream Loss-tolerant Authentication)와 전자서명에 대하여 기술한다.

2.3.1 TESLA[18]

TESLA는 비밀키 암호방식과 키해시를 이용하여 네트워크 메시지 인증을 위해 고안된 기법이다. 메시지를 m, 송수신자가 공유한 비밀키를 k, 키해시를 $H_1()$라고 하자. 송신자는 (1)과 같이 h를 계산한 후 $M=(m,h)$을 전송한다.

$$h \leftarrow H_1(m,k) \tag{1}$$

통신 경로를 통해 수신자가 수신한 값을 $M'=(m',h')$이라고 했을 때, 식 (2)가 참일 때 메시지가 위변조되지 않았음(즉, $m=m'$)을 확인할 수 있다.

$$H_1(m', k) = h' \tag{2}$$

TESLA에서는 전송되는 메시지마다 다른 키를 사용한다. 송신자가 선택한 시드키(seed key) k_n으로부터 (3)과 같이 해시함수 $H_2()$ 를 이용하여 키체인을 생성한 다음 마지막 킷값인 루트키(root key) k_0을 공개한다.

$$k_i \leftarrow H_2(k_{i+1}), \qquad i = 0, \ldots, n-1 \tag{3}$$

메시지 인증을 위해 키를 사용할 때는 키체인 생성의 역순인 $k_1, \ldots, k_n$ 순서로 사용한다. 해시함수의 일방향성을 이용하여 공격자가 키를 예측하지 못하도록 하면서 수신자는 키의 검증이 가능하도록 하기 위한 것이다.

i번째 메시지 m_i에 사용된 키를 k_i라고 하고 m_i에 대한 키해시값을 $h_i = H_1(m_i, k_i)$라고 했을 때 전송 메시지 M_i를 (4)와 같이 생성한다 (그림 2 참조).

$$M_i \leftarrow (m_i, h_i, k_{i-1}) \tag{4}$$

수신자는 M'_i을 수신할 때까지 m'_{i-1}의 인증을 미루는 "지연 인증(delayed authentication)"이다. M'_i로부터 k'_{i-1}를 수신하면 이전에 수신한 킷값 k_{i-2}를 이용하여 (5)와 같은 방법으로 k'_{i-1}을 먼저 검증($k'_{i-1} = k_{i-1}$임을 확인)한 후, (6)과 같이 m'_{i-1}을 인증($m'_{i-1} = m_{i-1}$임을 확인)한다.

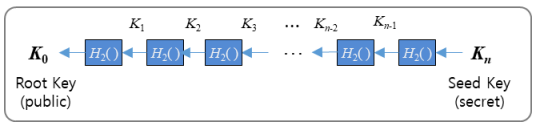


그림 1. 키체인 생성
Fig. 1. Key chain generation

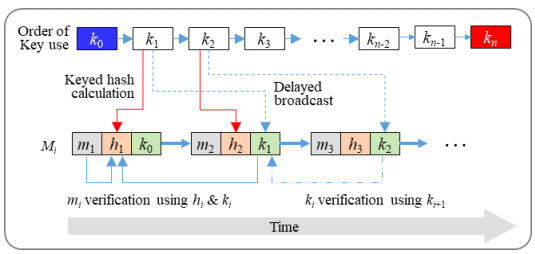


그림 2. 지연된 메시지 인증
Fig. 2. Delayed message authentication

$$k'_{i-2} = H_2(k_{i-1}) \ ? \tag{5}$$

$$h'_{i-1} = H_1(m'_{i-1}, k_{i-1}) \ ? \tag{6}$$

k_n까지 모든 키가 소진되면 새로운 키체인을 생성하여 사용한다.

2.3.2 디지털서명(Digital signature)

공개키 암호는 디지털서명에 사용될 수 있다. 메시지를 m, 서명 알고리즘을 $S()$, 서명 알고리즘을 $V()$라고 할 때, m에 대한 서명값 Sig는 서명자의 개인키 RK를 이용하여 (7)과 같이 계산된다.

$$Sig \leftarrow S(m, RK) \tag{7}$$

(m, Sig)를 수신한 개체는 서명자의 개인키 PK를 이용하여 (8)과 같은 방법으로 계산하고, 그 결과값에 따라 서명의 유효성을 검증한다.

$$V(m, PK, Sig) = Success \ ? \tag{8}$$

공개키 암호를 이용한 대표적인 디지털서명 알고리즘으로는 RSA[12], DSS(Digital Signature Standard)[19], ECDSA(Elliptic Curve Digital Signature Algorithm)[20], KCDSA(Korean Certificated-based Digital Signature Algorithm)[21] 등이 있다. 안전성의 척도인 보안 강도는 키의 길이가 길수록 향상된다.

Ⅲ. 위성항법시스템 인증 현황

3.1 국내외 위성항법시스템 운영 현황

위성항법시스템은 범지구를 커버리지로 하는 GNSS(Global Navigation Satellite System)와 지역을

표 1. 국내외 위성항법시스템
Table 1. Satellite navigation system

Nation	System	Coverage	Number of Satellites (Design/ Operating)	Authen-tication
USA	GPS	Global	24/31	△
EU	Galileo	Global	30/22	○
Russia	GLONASS	Global	24/23	×
China	BeiDou	Global	35/33	△
Japan	QZSS	Regional	7/4	△
India	NavIC	Regional	7/7	×
Korea	KPS	Regional	8/0	-

커버리지로 하는 RNSS(Regional Navigation Satellite System)로 구분된다. 현재 운용되거나 운영 예정인 위성항법시스템은 표 1과 같다. 유럽 연합의 Galileo에 대한 인증 서비스 테스트가 완료 되었으며, 미국의 GPS, 중국의 BeiDou, 일본의 QZSS에 대한 인증 연구도 진행되고 있다.

3.2 위성항법시스템 인증 기술

3.2.1 Galileo[8]

E1B 채널에서 전송되는 I/NAV 메시지 타입에 대하여 인증 서비스를 시범운용하고 있다. 위성항법 데이터는 프레임 단위로 반복 전송되는데, 1개의 프레임은 24개의 서브프레임으로 구성되고, 1개의 서브프레임은 15개의 페이지로 구성되며, 125bps로 2초에 걸쳐 전송된다.

Galileo에서는 페이지에 대하여 TESLA 방식을 이용하여 인증한다. 사용되는 해시함수와 루트키를 안전하게 배포하기 위한 디지털서명 방식은 선택할 수 있다.

홀수 페이지에 있는 40비트의 예비 필드에 페이지에 대한 인증 정보를 실어 보낸다. 인증 정보는 그림 3과 같이 15개 이상의 페이지에 걸쳐 페이지당 40비트씩 분할되어 30초 이상의 기간 동안 전송된다. HKROOT에 있는 정보는 TESLA 루트키에 대한 정보이고, MACK는 항법메시지 인증정보와 사용된 키를 포함한다.

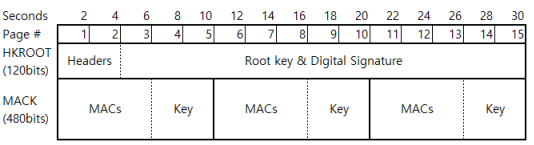


그림 3. Galileo 인증 데이터 구조
Fig. 3. Galileo authentication data layout

3.2.2 GPS[5,9,22,23]

CNAV-2 L1C 채널에서 전송되는 항법메시지와 확산코드(spreading code)에 대한 인증 방식으로서 Chimera (Chips Message Robust Authentication)가 제시되었다. L1C 메시지는 3개의 서브프레임으로 구성되며 18초에 걸쳐 100bps로 전송된다. 1번 서브프레임에는 TOI (Time Of Interval), 2번 서브프레임에는 시간과 ephemeris 정보, 3번 서브프레임에는 10개의 서로 다른 페이지가 전송된다. 전송되는 페이지의 순서는 가변적이며, 180초에 걸쳐 전송되는 10개의 페이지 전송 주기를 Chimera epoch라고 한다.

항법메시지 인증에는 224비트 ECDSA (ECDSA-224) 디지털서명을 이용하여 Chimera epoch를 주기로 10개 메시지에 대한 1개의 448비트 서명값을 전송한다. 그림 4에서와 같이 서명은 3번 서브프레임의 페이지 8과 9에 나누어 전송된다. 8번 페이지는 임의의 위치에서 전송될 수 있으며 9번 페이지는 마지막 10번째에 전송된다.

확산 코드에 대한 인증은 확산코드에 삽입된 마커를 검증함으로써 이루어진다. 마커의 삽입위치와 값에 대한 정보는 저속채널과 고속채널로 지원된다. 저속채널은 항법데이터 서명으로부터 도출되고 고속채널에서는 인터넷 등을 통해 별도로 지원된다. Chimera는 전자서명을 이용하기 때문에 서명 생성 및 검증 시간이 상대적으로 오래 걸리고 전송량도 많으며 인증 주기가 매우 길다.

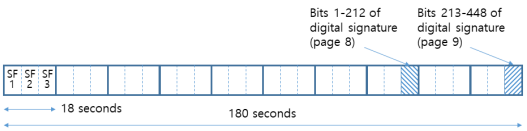


그림 4. GPS 디지털 서명 포맷
Fig. 4. GPS Digital signature format

3.2.3 BeiDou

BeiDou 위성 시스템은 정지궤도, 중궤도, 경사궤도 위성들로 구성된다. 중궤도와 경사궤도(IGSO) 위성은 50 bps로 전송되는 D1 메시지 포맷을 사용하고, 정지궤도 위성은 500 bps로 전송되는 D2 메시지 포맷을 사용한다. D1 메시지는 24개 메인프레임이 12분 동안 전송되는데, 각 메인프레임은 5개의 서브프레임으로 구성된다[24].

Zhijun Wu 등은[25]에서는 ECDSA를 이용하여 항법데이타에 대한 전자서명값을 생성하고 특정 메인프레임에 실어보낸다. D1 메시지에서는 5개 서브프레임 중 1-3번 서브프레임에 항법데이터가 전송된다. 그림 5와 같이 1-12번 메인프레임의 항법데이터에 대한 서명은 11-12 메인프레임의 5번 서브프레임에 반씩 나누어 실어 보낸다. 마찬가지로 13-24번 메인 프레임의

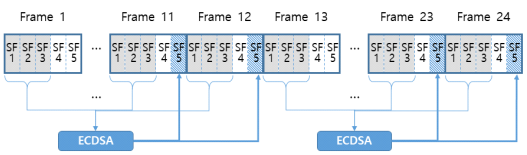


그림 5. BeiDou D1 메시지에 대한 BD-NMA 인증방식
Fig. 5. BD-NMA for D1 message

항법데이터에 대한 서명은 23번과 24번 메인프레임에 실어 보낸다.

D1 포맷에서는 24개의 메인프레임마다 3개의 항법데이터 서브프레임이 전송되는데, 이 방식에서는 11번과 23번 항법데이터에 대해서만 인증하고 있다. NIST는 2030년까지의 사용을 전제로 ECDSA-224 전자서명을 권고하고 있다. 저자들은 ECDSA의 키 길이를 지정하지 않았으나 D1에서는 서명값이 전송될 여분이 공간이 356비트이기 때문에 448비트가 소요되는 ECDSA-224을 사용할 수 없다.

Zhijun Wu 등은[26] 500bps로 전송되는 D2 메시지에 대한 인증 방식을 제안하였다. D2 메인프레임은 5개의 서브프레임으로 구성되며, 항법 데이터(BNI: Basic Navigation Information)는 10개의 페이지로 구성되는데 10개의 1번 서브프레임에 나뉘어 30초에 걸쳐 전송된다. 이 10개의 서브프레임을 1개의 그룹이라고 하며 인증 단위가 된다. 이 방식에서는 미코닝(Meaconing)을 방지하기 위한 신호레벨의 인증방식도 포함된다. 이를 위해 그림 6에서와 같이 10개의 서브프레임 1에 포함된 SOW(Second of Week) 정보를 가지고 그룹 인증정보(Group Auth)와 페이지 인증정보(Page Auth)를 생성한다. 그룹 인증정보와 확산 스펙트럼 변조를 위한 GPSSS(Generator Polynomial of Spectrum Spreading Sequence)를 128비트 키를 사용하는 SM4 대칭키 암호 알고리즘으로 암호화하여 128비트 암호문을 10개의 서브프레임 1번의 여분 공간에 나누어 삽입한다. 인증 대상인 항법 데이터 BNI (Basic Navigation Information)에 대하여 SM2 알고리즘으로 생성한 전자서명과 페이지 인증정보를 GPSSS 방식으로 변조하여 75비트의 SSI를 생성한 후 서브프레임 1과 2 사이에 전송한다. SM2는 256비트 공개키와 128비트 개인키를 사용한다.

수신자는 3단계에 걸쳐 인증을 수행한다. 첫 번째

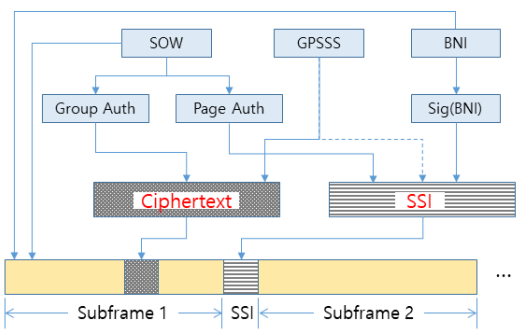


그림 6. BeiDou II의 D2 메시지에 대한 인증방식
Fig. 6. BD II authentication for D2 message

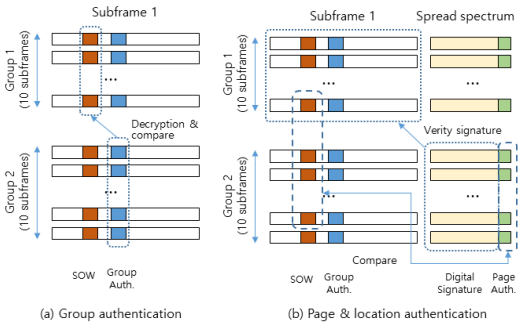


그림 7. BeiDou II 인증
Fig. 7. BeiDou II authentication

로 이전 서브프레임 1의 10개 페이지에서 SOW로부터 그룹 인증정보를 생성하여 다음 10개 페이지에서 수신한 암호문을 복호화여 추출한 SOW의 그룹 인증정보와 동일한지를 비교함으로써 그룹 시간정보를 인증한다 (그림 7 참조).

두 번째로 암호문으로부터 추출한 GPSSS 정보를 가지고 SSI를 변조하여 서명과 페이지 인증정보를 추출하여 이전 페이지의 SOW에서 생성한 페이지 인증정보와 동일한지 검증한다. 마지막으로, 이전 그룹에 속한 10개 서브프레임의 BNI에 대한 SM2 서명을 SSI에서 추출하여 검증한다.

이 방식은 신호레벨의 보안 방식을 도입하여 미코닝을 방지하도록 고안되었다는 장점이 있으나, 500bps의 매우 빠른 채널을 이용함에도 불구하고 초기 항법데이터 인증에 1분 이상의 시간이 소요된다. 항법데이터의 서명으로부터 무결성을 검증하면서도 비밀값이 아닌 SOW를 암호화하여 전송하고 이를 통해 인증하도록 하는 등 중복되고 복잡한 인증 절차를 거친다. 또한 이 방식에서 사용하는 비밀키 방식은 상호간에 키를 공유해야 하는 큰 난제를 가지고 있으나 이에 대한 해결책으로서 모든 수신기가 마스터키를 보유하도록 하고 이를 이용하여 SMS를 통해 업데이트한다. 이 마스터키는 대중에게는 알려지지 않고 수신기 제조업체만 보유하도록 하는 암호 알고리즘에 의해 보호한다고 한다. 이 방법에는 여러 가지 문제가 제기될 수 있다. 우선 암호알고리즘을 공개적하여 안전성을 검증하고 최소의 키만 비밀로 유지하도록 하고 있는 일반적 방식을 따르지 않고 있다. 또한 수신기를 소유한 공격자도 마스터키를 통해 비밀키를 알아낼 수 있으며, 모든 제조업체가 암호알고리즘을 비밀로 유지하는 것은 현실적으로 매우 어렵다.

3.2.4 QZSS

Dinesh Manandhar 등은[27] QZSS 뿐만 아니라 수신가능한 모든 위성으로부터의 항법 데이터를 인증할 수 있는 방법을 제안하였다. 그림 8에서와 같이 ADC(Authentication Data Center)에서 QZSS L1C/A, GPS L1C/A 및 갈릴레오 E1B 신호로부터 항법 데이터를 수신하고 추출한 항법데이터에 대한 전자서명을 생성한다. 생성된 전자서명은 QZSS 항법데이터 메시지 포맷의 항법데이터 대신 들어간다. 이렇게 생성된 신호를 위성으로 업로드하면 QZSS 위성은 L1S 신호를 통해 사용자들에게 브로드캐스트한다.

L1S 수신자는 PRN ID로부터 인증 대상인 위성을 식별하고 수신한 항법데이터에 대한 전자서명을 검증한다.

이 방식에서는 다양한 위성으로부터의 항법데이터를 인증할 수 있다는 장점이 있지만 별도의 채널이 필요하며 낮은 안전도를 가진다. 적용할 전자서명 알고리즘이 제시되고 있지 않지만 전자서명에 할당된 공간이 192비트로서, 적용할 수 있는 전자서명 알고리즘이 지극히 제한적이며 짧은 길이로 인해 NIST에서 권고하는 안전성을 만족하지 못한다.

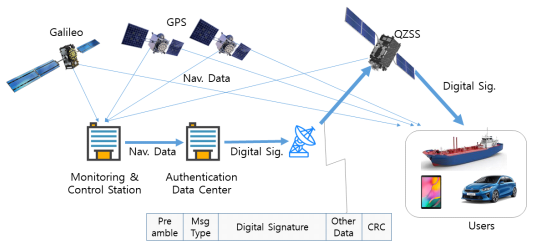


그림 8. QZSS 인증 구조
Fig. 8. QZSS authentication architecture

Ⅳ. 제안 인증 방안

4.1 항법메시지 구조

한국전자통신연구원에서는 차세대 위성으로서 신호생성기 및 위성항법메시지 구조를 설계하였다[28,9]. 본 연구에서는 이 메시지 구조에 기반하여 인증 방안을 연구하였으며, 개괄적 설계를[30] 발표하였다. 본 논문에서는 KPS에 적용할 수 있는 인증방식의 상세 설계와 성능 분석 및 타 방식과의 비교, 그리고 키갱신 설계를 추가하였다.

1개의 I/NAV 메시지는 4개의 서브프레임으로 구성되고, 각 서브프레임은 그림 9와 같이 동기화를 위

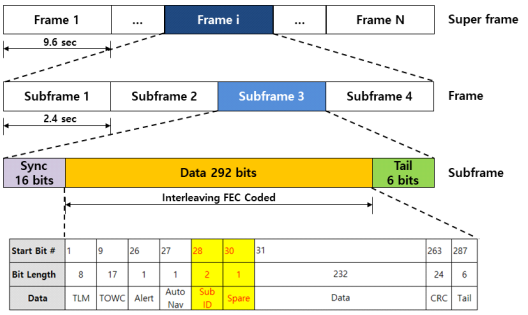


그림 9. 한국전자통신연구원이 제안한 메시지 포맷
Fig. 9. Navigation message format designed by ETRI

한 16비트, 항법메시지 292비트, 그리고 테일 6비트로 구성되어 2.4초에 걸쳐 전송된다. 음영처리된 28-30번 비트는 인증 메시지를 위해 변경될 필드이며 Data 필드에 인증 데이터가 삽입될 것이다.

서브프레임은 2가지 타입을 가진다. 서브프레임 1과 2(SF1, SF2)는 NMS (Navigation Message Subframe) 타입으로서 항법데이터로 구성된다. 서브프레임 3과 4(SF3, SF4)는 MTS (Message Typed Subframe) 타입으로서 보정 데이터로 구성된다. 292비트 항법메시지에는 앞부분에 TLM, TOWC 등 30비트 정보와 CRC 24비트가 마지막에 고정적으로 포함된다(그림 10 참조).

4.2 항법메시지 인증 방안 제안

4.2.1 설계 개요

(1) 설계 기준

암호 기술을 이용하여 인증기능을 적용하기 위해서는 항법데이터에 비하여 상대적으로 많은 비트의 인증정보가 전송되어야 한다. 본 연구에서는 안전성과 가용성의 적절한 균형을 유지하도록 다음과 같은 기준으로 설계하였다.

- 평균 최초 인증시간과 평균 연속 인증시간을 최소화
- 위성과 단말기의 낮은 성능을 고려하여 계산시간 최소화
- 인터넷 등 별도 채널 사용 제한
- 성능을 위하여 보정 데이터를 제외한 항법 데이터만을 인증 대상으로 한정
- 설계된 항법메시지 구조를 유지
- KISA 및 NIST 안전도 권고사항 준수

(2) 적용 암호 기술

암호 알고리즘의 수행시간을 분석하기 위하여 파이썬프로그램에서 암호 모듈을 이용하여 비교하였다. 표 2와 같이 메시지 인증에 전자 서명을 사용할 경우, MAC (Message Authentication Code)에 비해 계산 시간도 길어지고, 긴 비트 길이로 인하여 전송량이 많아지기 때문에 항법메시지 수신 시간 및 인증 시간이 더 많이 지연된다.

따라서 본 연구에서는 키해시 함수를 이용한 TESLA 방식을 적용하였다. 단, 키체인의 루트키 공유를 위한 인증에는 전자서명을 이용한다. TESLA 키체인은 1일 주기로 갱신할 수 있는 길이로 생성한다. 지연시간을 최소화하기 위하여 보정 정보를 제외한 항법메시지 서브프레임만을 인증 대상으로 하며, 이 메시지에 대한 인증은 별도의 인증 서브프레임(*KAS*: Key and Authentication Subframe)을 설계하여 실어 보낸다. 적용된 설계 요소는 표 3과 같다.

표 2. 암호 알고리즘 분석 결과
Table 2. Cryptographic algorithm analysis result

Algorithm		Key Len. (bits)	Len. of Result (bits)	100,000 Execution Time (sec)	Comp. with SHA1 (times)
SHA1		-	160	0.067	1.00
HMAC-SHA1		160	160	0.137	2.04
ECDSA -224	Sign	224	448	37.637	561.75
	Verify	224	448	33.045	493.21

표 3. 적용 설계 요소
Table 3. Applied design elements

Design Element	Applied Method	Remark
Auth. Target	Subgrame 1, 2	NMS
Auth. Data Transmission	Separated New Auth. Subframe	8 Types
Auth. Algorithm	HMAC-SHA1	160 bits
Auth. Key Sharing	TESLA	1 Day Cycle
Root Key Auth.	ECDSA-224	448 Bits

4.2.2 인증 메시지 설계

(1) *KAS* 구조

표 4에서 음영 처리된 서브프레임은 인증 서브프레임(*KAS*)이다(SF Id 1, 2). *MTS*(SF Id 3, 4)는 3번의 *NMS* 전송 후에 1번 전송된다. 2개의 *NMS* 와 *MTS* 전송 후에는 1개의 *KAS*가 전송된다(SF Id 17-24). 인증정보는 8가지의 *KAS*에 실어보낸다. 따라서 24개의 서브프레임을 주기로 인증 서브프레임이 반복 전송된다.

표 4. 서브프레임 전송 순서
Table 4. Subframe transmission order

Seq	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
SF Id	01	02	17	01	02	18	01	02	19	03	04	20	01	02	21	01	02	22	01	02	23	03	04	24

*KAS*는 루트키 인증정보 섹션(*RK_Sec*)과 항법데이터 인증 정보 섹션(*MAC_Sec*)으로 구성된다. *RK_Sec*은 항법데이터 인증에 사용된 TESLA 키체인의 루트키에 대한 인증 정보로서, 키체인 식별자(*Ch_Id*1), 루트키(*Root_Key*), 루트키에 대한 서명값(*Sig*1,*Sig*2) 그리고 서명 검증에 사용될 공개키의 식별자(*PK_Id*)로 구성된다. *MAC_Sec*은 항법데이터에 대한 인증 정보로서, 사용 중인 키체인 식별자(*Ch_Id*2), 키체인 내에서 항법메시지 인증에 사용된 키의 일련번호(*Ch_Seq*), 그리고 항법메시지 인증값(*MAC*), 그리고 이전 항법메시지 인증에 사용된 킷값(*MAC_Key*)이 포함된다. *MAC_Sec*은 매 항법메시지 서브프레임 후에 전송한다. 처음 혹은 오랜만에 수신 장치를 사용하는 경우에도 언제든 루트키를 인증할 수 있도록 *RK_Sec*도 주기적으로 전송한다.

(2) *KAS* 상세 설계

*RK_Sec*와 *MAC_Sec*에 포함된 각 필드의 비트 길이를 설계하고, 그에 따라 *KAS*의 메시지 구조를 설계한다.

하나의 TESLA 키체인은 1일 동안 사용할 수 있도록 하며 24시간마다 새로운 키체인을 생성한다. 사용하는 키체인을 식별자인 *Ch_Id*1과 *Ch_Id*2는 일주일을 주기로 재사용할 수 있도록 3비트(1-7)로 정한다. 공개키 식별자인 *PK_Id*는 8비트를 할당하여 255개(1-255)의 공개키를 구분하여 사용할 수 있도록 한다(식별자 0은 사용하지 않는다). 항법데이터에 대한 인증값(*MAC*)은 전송량을 줄이기 위하여 항법 메시지 1, 2에 HMAC-SHA1을 적용한 결과값 160비트 중 23비트를 사용한다. 23비트를 추출하는 방법은 상위 23비트를 사용할 수도 있고, OTP(One Time Password)에서 제안하는 방식 등을 사용할 수 있다[31]. 루트키와 인증 키는 SHA1을 사용하므로 160비트이다(표 6 참조). 해당 *Ch_Id*2의 키체인에서 적용한 키를 식별하기 위한 일련번호 *Ch_Seq*의 길이는 표 5와 같이 계산된다. 하나의 키체인은 24시간(86,400초) 동안 사용할 수 있어야 한다. 하나의 서브프레임 전송

시간이 2.4초이므로 1인증 주기인 24 서브프레임의 전송은 57.6초 소요된다. 따라서 하루에 1,500번의 인증 주기가 반복된다. 한 인증 주기에는 6개의 인증키가 필요하므로 하루에 소요되는 키는 9,000개가 되어, 이들 키를 식별하기 위한 *Ch_Seq*의 최소 길이는 14비트이다. 표 5에 요구되는 비트 길이를 요약하였다.

표 7과 같이 *KAS*는 *RK_Sec*과 *MAC_Sec*으로 이루어진다. 하나의 루트키에 대한 640비트 인증값 *RK_Sec*은 8개 *KAS*에 나뉘어 전송된다. *NMS* 다음에 전송되는 17-19번과 21-23번 *KAS*에는 바로 앞에 전송된 2개의 *NMS*에 대한 인증값인 *MAC_Sec*과 루트키 인증정보인 *RK_Sec*이 포함된다. *MTS*는 인증 대상이 아니기 때문에 *MTS* 다음에 전송되는 20과 24번 *KAS*에는 *RK_Sec*만 포함된다.

8개의 *KAS*의 상세 구조는 그림 10과 같다. *KAS*를 식별하기 위하여 Sub ID 필드가 5비트로 변경되고

표 5. *Ch_Seq*의 비트 길이 계산
Table 5. Calculation of the *Ch_Seq* bit length

Term	Formula	Result
seconds / day	24 × 60 × 60	86,400 sec
cycles / day	86,400 / 57.6	1,500 cycles
keys / cycle	6	6 keys
keys / day	1,500 × 6	9,000 keys
Ch_Seq length	$\lceil \log_2(9{,}000) \rceil$	14 bits

표 6. *KAS*의 데이터 필드 길이
Table 6. Bit length of data fields in *KAS*

Section	Message Field	Bit Length	Total Bit Len.
RK_SEC	*PK_Id*	8	640
	*Ch_Id*1	3 * 8 = 24	
	Root_Key	160	
	*Sig*1, *Sig*2	422 * 2 = 448	
MAC_SEC	*Ch_Id*2	3	200
	Ch_Seq	14	
	MAC	23	
	MAC_Key	160	

표 7. *KAS* 비트 구성
Table 7. *KAS* bit configuration

SF Id.	17	18	19	20	21	22	23	24	Total
RK_Sec	30	30	30	230	30	30	30	230	640
MAC_Sec	200	200	200		200	200	200		

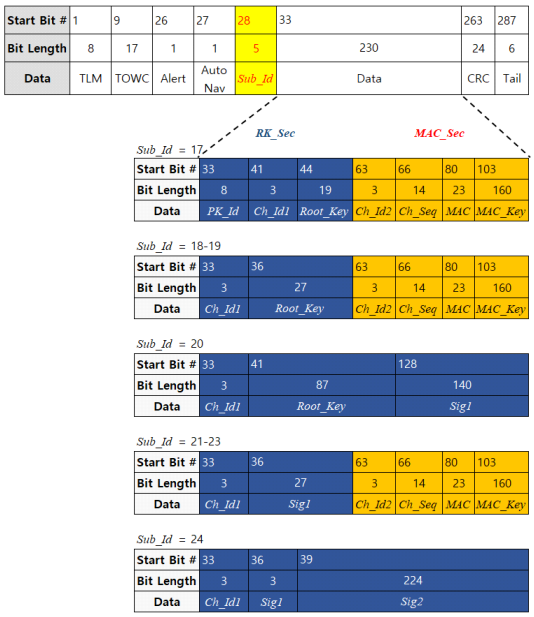


그림 10. *KAS*의 필드 구성
Fig. 10. Field composition of *KAS*

Spare 필드가 삭제되었으며 이에 따라 Data 필드가 230비트로 조정되었다. *NMS*와 *MTS*에서도 Data 필드에 포함된 2비트의 Space 비트를 Sub Id로 이용한다.

4.2.3 인증 방법

그림 11은 *KAS*에 있는 데이터의 관계를 보여준다. 예로, 6번째로 전송된 18번 *KAS*의 *MAC_Sec*에 있는 MAC은 4, 5번째로 전송된 *NMS*에 대한 HMAC-SHA1 값이다. *MAC_Key*는 3번째로 전송된 17번 *KAS*의 MAC값(1, 2번째 *NMS*에 대한 MAC)을 계산하는데 사용된 킷값이다. 따라서 18번 *KAS*를 수신할 때까지 17번 서브프레임의 MAC값에 대한 검증 과정이 지연된다.

이 절에서는 앞에서 기술한 방법에 따라 구성된 메

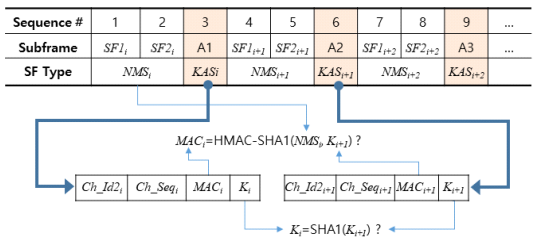


그림 11. *NMS* 인증 데이터 관계
Fig. 11. *NMS* authentication data relationship

시지가 브로드캐스트되었을 때 수신기의 인증 방법에 대해 기술한다.

수신기는 초기화 단계에서 공개된 위성 공개키를 획득할 수 있다고 전제한다. 수신기가 NMS를 인증하기 위해서는 다음 단계를 거친다.

(1) 루트키 인증 단계

단말기를 처음 사용하거나 오랫동안 사용하지 않다가 다시 사용하는 경우에는 현재 사용하고 있는 키체인의 루트키를 인증해야 한다. 루트키 인증은 키체인이 다 소진될 때까지 한번만 수행하면 된다. TESLA 키체인은 해시함수 SHA1을 이용하여 생성한다. 루트키를 인증하기 위해서는 8개의 KAS를 모두 수신한 후 RK_Sec에서 각 항목들을 추출한다. 추출한 PK_Id로부터 루트키에 대한 서명을 검증하기 위한 공개키를 식별한다. 루트키가 포함된 키체인의 식별자인 Ch_Id1를 저장한다. 루트키 $Root_Key$는 항법메시지 인증에 사용될 MAC_Key_0로 사용된다. 프로시져 1과 같이 $Root_Key$에 대한 서명값 $< Sig_1, Sig_2 >$을 공개키로 검증함으로써 루트키를 인증한다.

프로시져 1. 루트키 인증 절차
Procedure 1. Procedure for Root key authentication

1. Receive 8 KASs
2. Extract fields from RK_SEC
3. Identify public key from PK_Id
4. Save Ch_Id1
5. Extract & Save Root Key $Root\_Key$ as MAC_Key_0
6. Verify $Root\_Key$ using digital signature $< Sig_1, Sig_2 >$ and public key

(2) 항법데이터 인증 단계

루트키에 대한 인증이 이루어지고 나면, 항법데이터에 대한 인증 단계를 수행할 수 있다. 항법데이터 NMS에 대한 인증은 항법데이터 다음에 전송되는 KAS를 통해 이루어지기 때문에 그 때까지 수신한 NMS를 저장한다. 편의상 i번째 수신한 두 개의 NMS 서브프레임을 NMS_i라고 표기하고 이에 대응되는 KAS를 KAS_i라고 표기한다. 또한 KAS_i에 포함된 필드에는 Ch_Id1_i와 같이 동일한 첨자를 사용한다. 저장된 NMS에 대한 인증을 하기 위해서는 먼저 인증에 사용되는 키 값을 인증해야 한다. 프로시져 2와 같이 사용 중인 키체인을 확인하기 위해서 MAC_Sec에서 추출한 Ch_Id2_i가 KAS의 RK_Sec에서 수신한 Ch_Id1과 동일한지 확인한다. 또한 키체인이 변경되지 않았음을 확인하기 위해 이전에 수신한 Ch_Id2_{i-1}과 동일한지 확인하고 $Ch_Seq_i = Ch_Seq_{i-1} + 1$과 같이 키체인의 일련번호를 확인한다. 만약 중간에 수신하지 못한 KAS가 있다면 +1이 아니라 $+n(n \geq 2)$가 될 수 있으나, 여기서는 편의상 수신하지 못한 KAS가 없다고 가정하고 기술한다. 마지막으로 현재 수신한 인증키 MAC_Key_i를 해시($SHA1()$)하면 이전에 수신한 인증키 MAC_Key_{i-1}와 같아지는지 확인함으로써 인증키에 대한 인증을 마친다.

프로시져 2. 인증키 인증 절차
Procedure 2. Procedure for auth. key authentication

1, Save NMS_{i-1} and NMS_i
2. Extract Fields from MAC_Sec in KAS_i
3. $Ch\_Id_1 = Ch\_Id2_i$ && $Ch\_Id2_{i-1} = Ch\_Id2_i$?
4. $Ch\_Seq_i = Ch\_Seq_{i-1} + 1$?
5. $MAC\_Key_{i-1} = SHA1(MAC\_Key_i)$?

다음은 인증된 인증키로 저장된 이전 NMS를 인증하는 단계이다. 프로시져 3과 같이 현재의 KAS에서 추출하여 1단계에서 인증된 인증키 MAC_Key_i로 이전 항법 메시지 NMS_{i-1}을 검증하기 위하여 키해시 값을 계산한다. 계산값이 이전 KAS에 포함된 값인 MAC_{i-1}과 같으면 이전 항법데이터에 대한 인증이 완료된다.

프로시져 3. 항법데이터 인증 절차
Procedure 3. Procedure for Nav. data authentication

1. $HMAC\_SHA1(NMS_{i-1}, MAC\_Key_i) = MAC_{i-1}$?
2. Save Ch_Id2_i, Ch_Seq_i and MAC_Key_i
3. Delete NMS_{i-1}

4.2.4 루트키 및 키체인 생성 방법

인증의 안전성은 인증값을 예측할 수 없다는 것이 전제되어야 한다. 인증에 사용되는 키의 노출이나 예측은 모든 인증값의 노출을 의미하기 때문에 매우 중요하다. 비밀로 유지해야 하는 키의 개수는 적을수록 좋다. 따라서 지속적으로 갱신해야 하는 루트키를 그림 12와 같이 생성할 수 있다. 인증용 키체인 생성에 사용되는 해시함수를 $H()$라고 할 때, 또 다른 해시함수 $F()$를 선정하여 키체인의 시드키를 생성하는데 사

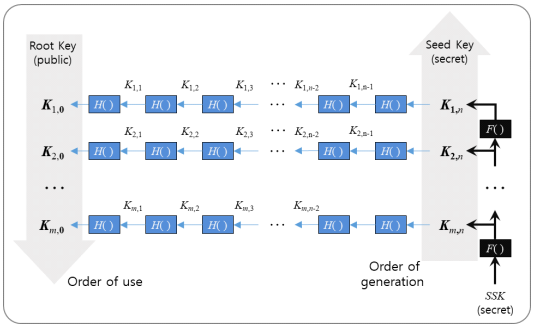


그림 12. 루트키 생성 방법
Fig. 12. Root key generation method

용된다. 센터에서는 시드키의 시드(SSK: Seed of Seed Key)를 비밀로 유지하고 이 값으로부터 $F()$를 통하여 시드키 체인 $K_{m,n}, K_{m-1,n}, \cdots, K_{2,n}, K_{1,n}$을 생성한 후, $K_{i,n}$부터 시작해서 생성의 역순으로 시드키로 사용하여 키체인을 생성한다. 키를 생성하고 관리하는 제어센터에서는 하나의 SSK만 유지하면 된다.

이 방법을 사용할 경우 루트키 인증 시간을 크게 줄일 수 있다. 시드키 체인의 길이는 정책적으로 설정할 수 있다. $F()$가 160비트 해시함수라면 2^{160}개의 서로 다른 값을 생성해 내므로 매우 긴 체인이 가능하다. 예로 체인의 길이를 365로 정한다면, 인증 키체인의 사용기간이 1일이므로 1년을 사용할 수 있다. 동일한 시드키 체인이 사용되는 동안에는 사용이 만료된 최종 시드키를 저장하고 $F()$를 이용하여 다음 시드키를 인증한다면, 전자서명이 포함된 서브프레임을 수신하지 않아도 되므로 루트키 인증에 소요되는 시간을 최대 1/2로 줄일 수 있다. 단, 처음 수신기를 사용하거나 SSK가 변경되는 경우에는 전자서명을 확인해야만 한다.

V. 분석 및 비교

5.1 성능 분석

5.1.1 성능 척도

기존의 인증방법과 성능을 비교하기 위한 4가지 척도로서 인증시간에 관련된 TFAF와 TBA, 인증 적용에 따른 서비스 지연에 관련된 TFUD와 TBUD를 도입하였다. 위성이나 수신기의 인증값 계산이나 검증에 소요되는 시간은 위성으로부터의 데이터 전송시간에 비해 미미하기 때문에 계산값에 반영하지 않았다.

(1) TFAF (Time to First Authentication Fix)

위성항법 데이터를 인증하기 위해서는 항법데이터와 인증정보뿐만 아니라, 인증정보 검증에 필요한 데이터의 수신이 필요하다. 즉, 공개키나 MAC 키 등을 수신해야 한다. 처음 수신기 전원을 켰을 때나 오랜 시간 후 수신기를 켰을 때는 이러한 사전정보 수신 시간이 필요 곧바로 인증이 불가능할 수 있다. TFAF는 이러한 시간을 포함하여 첫 인증된 위성항법데이터를 수신하기까지의 시간을 의미한다.

(2) TBA (Time Between Authentication)

위성항법데이터를 인증한 후, 다음 인증까지의 시간 간격(즉, 연속 인증 시간)을 의미한다. TFAF가 위성항법 데이터 사용자가 기다려야 하는 최악의 시간이라면, TBA는 실질적인 인증 시간이라고 볼 수 있다. 따라서 TFAF 보다 짧다.

(3) TFUD (Time to First Unauthenticated Data)

이 척도는 인증 정보가 추가된 구조에서 인증 기능을 사용하지 않고자 할 때의 최초 위성항법 데이터 수신 시간을 의미한다. 인증정보 추가로 인해 발생하는 지연시간 평가에 사용된다.

(4) TBUD (Time Between Unauthenticated Data)

이 척도는 인증 정보가 추가된 구조에서 인증 기능을 사용하지 않고자 할 때, 연속적인 위성항법 데이터 수신시간 간격을 의미한다. 인증정보 추가로 인하여 발생하는 지연시간 평가에 사용된다.

5.1.2 인증 서브프레임 배치에 따른 성능 분석

항법데이터를 인증하려면 먼저 공개키를 통하여 루트키를 인증해야만 해야 한다. 사용자 수신기를 오랜만에 켜거나 처음 사용하는 경우에는 최대 24개 서브프레임을 수신해야만 하므로 57.6초만에 루트키를 인증할 수 있다. KPS는 정지궤도 위성 3기와 경사궤도 위성 5기로 이루어져 있고, 이 중 4기 이상의 위성으로부터 데이터를 수신할 때 위치를 계산할 수 있다. 수신자의 위치에 따라 가시 위성을 예측할 수는 없지만, 그림 13과 같이 인증 프레임을 배치하면 8개의 인증프레임 수신 시간을 줄일 수 있을 것이다. 사용자가 수신키를 켜는 시점에 따라 루트키 인증 시간, TFAF, TBA, TFUD, TBUD이 다르다. 그림 13의 하단부는 각각에 대하여 최소와 최대 시간을 가지는 항법데이터 수신 기간에 대한 예시이다.

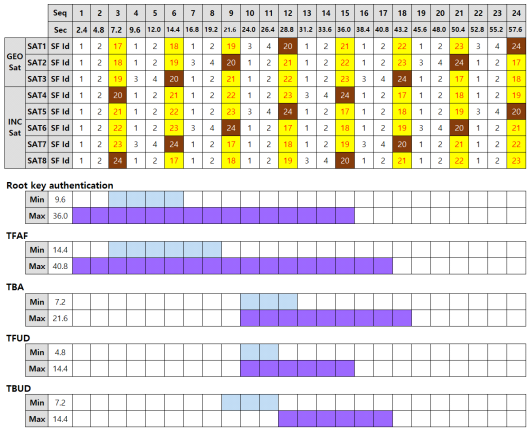


그림 13. 인증 서브프레임 배치 순서
Fig. 13. Ordering of authentication subframes

5.1.3 성능 비교

제안 방식의 성능 분석 항목에 대하여 Galileo, GPS 및 BeiDou II와 비교하였다. BeiDou D1을 대상으로 하는 방식은 항법메시지 전체를 인증하지 못하므로 비교에서 제외하였고, QZSS는 여러 위성에 대한 인증 방식이기 때문에 동등 비교가 불가능하여 제외하였다. 또한 제안 방식에서 *SSK*를 이용하는 경우는 성능이 향상되지만 동일한 *SSK*를 사용하는 기간일 때와 그렇지 않을 때가 다르기 때문에 비교하지 않았다.

Galileo는 키 길이와 MAC 개수 등이 가변적이어서 소요 시간도 가변적이다. 따라서 인증방식에 대한 테스트 규격을[9] 기준으로 비교하였다. GPS는 전자서명만을 이용하며, 공개키 수신을 위성을 사용하는 Slow Channel 방식과 인터넷을 사용하는 Fast Channel 방식으로 구분한다. 본 연구에서는 인터넷을 이용한 인증방식을 고려하지 않기 때문에 Slow Channel의 경우와 비교하였다. 표 8에 표시한 비교결과를 보면 제안한 방식이 다른 방식에 비해 모든 항목에서 우수한 것을 알 수 있다. 이는 인증 대상인 항법데이터의 범위와 전송 주기를 조정하고 인증 주기를 짧게 설계함으로써 나타난 결과이다.

표 8. 성능 비교 (단위: 초)
Table 8. Performance comparison (unit: seconds)

Criteria \ NSS		Proposed	Galileo	GPS	BeiDou
TFAF	Min	14.4	60.0	180.0	62.0
	Max	40.8	120.0	360.0	93.5
TBA	Min	7.2	30.0	180.0	31.5
	Max	21.6	60.0	180.0	31.5
TFUD	Min	4.8	30.0	30.0	30.0
	Max	14.4	60.0	60.0	60.0
TBUD	Min	7.2	30.0	30.0	30.0
	Max	14.4	30.0	30.0	30.0

5.2 안전성 분석

5.2.1 안전성 비교

안전도는 기본적으로 사용하는 키의 길이에 따라 평가할 수 있다. 사용할 수 있는 키의 길이에 대하여 Galileo, GPS 및 BeiDou와 비교하였다. Galileo는 본 연구의 제안방식과 유사한 구조를 가지고 있어 표 9에서 항법데이터 인증과 루트키 인증에 대하여 사용하는 키와 인증값의 비트 길이를 비교하였다. GPS는 전자 서명만을 사용하기 때문에 동등비교는 어려우나 전자서명값의 비트길이를 비교하였다.

제안한 방식에서는 인증을 위한 지연 시간의 최소화와 인증 성능을 우선적으로 고려하여 적용 암호기술과 안전도를 좌우하는 비트길이를 고정하였다. Galileo의 경우에는 옵션으로 다양하게 적용할 수 있다. 루트키 인증에 사용되는 전자서명은 선택에 따라 달라지기 때문에 그 최대 길이를 알 수 없다. BeiDou에서 사용하는 전자서명 알고리즘 SM2의 키길이는 안전성이 의존하는 개인키의 길이로 표시하였다. 표 9에서 보는 바와 같이 제안한 방식은 다른 방식에 비해 대부분 높은 안전도를 보여준다. 단, 항법 데이터의 안전도는 Galileo 방식과 비교했을 때 평균 정도의 안전도를 가진다. 이는 Galileo 방식에서 제공하는 키길이 옵션 대신 적절한 안전성과 성능 향상을 고려한 설계이기 때문이다.

표 9. 안전도 비교 (단위: 비트)
Table 9. Security comparison (unit: bits)

Auth. target	Item		Prop.	Galileo	GPS	BeiDou
Nav. data	MAC/ Enc key	Min	160	100	-	128
		Max	160	256	-	128
	MAC	Min	23	10	-	-
		Max	23	32	-	-
Root key	Root key	Min	448	448	-	-
		Max	448	> 448	-	-
	Digital Sig.	Min	448	448	448	128
		Max	448	> 448	448	128

Ⅵ. 결 론

4차산업혁명 등에서 다양한 영역으로 GNSS 서비스가 확대됨에 따라 위장 공격의 시도도 많아질 것으로 예상된다. 이러한 공격으로부터 피해를 줄이기 위해 인증의 중요성이 더욱 중요해 졌다. 유럽의 Galileo를 시작으로 GPS, Beidou, QZSS도 인증 서비스에 대한 연구가 진행되고 있으므로 2035년 서비스 될 우리나라 KPS의 민간 항법데이터에 대해서도 인증서비스를 고려해야 할 것이다.

각 위성시스템은 서로 다른 환경과 메시지 구조를 가지고 있어 그 구조에 맞는 인증 방식을 사용해야 한다. 본 논문에서는 각국에의 위성에 대한 대표적인 인증방식에 대해 분석하고 한국전자통신연구원에서 설계한 민간위성항법메시지에 대하여 KPS에 적용할 수 있는 인증 방식을 제안하였다. 제안한 방식은 인증으로 인한 지연시간을 최소화하고 계산량을 줄일 수 있도록 설계한 인증용 서브프레임을 추가함으로써, 우수한 안전성을 제공하면서도 다른 시스템에 비하여 짧은 인증시간을 보여주고 있다. 만약 SSK를 이용한다면 루트키 인증으로 인한 지연을 더 줄일 수 있다. 향후, 단말기 성능, 타 위성과의 연계성, 안전성 등을 고려하여 세부적인 서비스 및 인증 정책이 수립된다면, 제안한 방안을 기반으로 더 정교하고 적절하게 확장된 설계할 수 있을 것이다.

References

[1] Mordor Intelligence, "*GNSS Chip Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026)*" from https://www.mordorintelligence.com/industry-reports/gnss-chip-market.

[2] European GSA, "*Galileo High Accuracy Service (HAS)*," European GNSS Agency, 2020.

[3] KCC and KISA, "*Monthly Report for Geolocation Industry Trend: Satellite Navigation System and Broadcast RTK for the 4th Industrial Revolution*," Feb. 2021.

[4] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS Spoofing," *J. Field Robotics*, vol. 31, no. 4, pp. 617-636, 2014.

[5] D. A. Divis, "GPS spoofing experiment knocks ship off course," *Inside GNSS*, Jul. 2013.

[6] C. Günther, "A survey of spoofing and countermeasures," *J. Inst. Navig.*, vol. 61, no. 3, pp. 159-177, Sep. 2014.

[7] Chosun Biz, "*North Korea jamming attacks 4 times since 2010*," from https://biz.chosun.com/site/data/html_dir/2016/04/17/2016041702172.html?form=MY01SV&OCID=MY01SV.

[8] European Commission, "*Tests of Galileo OSNMA underway*," Feb. 2021, from https://ec.europa.eu/defence-industry-space/tests-galileo-osnma-underway-2021-02-11_en.

[9] Air Force Research Laboratory (AFRL) Space Vehicles Directorate, Advanced GPS Technology, "*Chips Message Robust Authentication (Chimera) Enhancement for the L1C Signal: Space Segment/User Segment Interface*," IS-AGT-100, Apr. 2019.

[10] NIST, *Advanced Encryption Standard*, FIPS Publication 197, 2001.

[11] H. J. Lee, et al., *The SEED Encryption Algorithm*, IETF RFC4269, 2005.

[12] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, pp. 120-126, 1978.

[13] N. Koblitz, "Elliptic curve cryptosystems," *Math. Computation*, vol. 48, no. 177, pp. 203-209, 1987.

[14] NIST, "*Secure Hash Standard*," FIPS Publication 180-1, Apr. 1995.

[15] NIST, "*Secure Hash Standard*," FIPS Publication 180-2, Aug. 2002.

[16] NIST, "*The Keyed-Hash Message Authentication Code (HMAC)*," FIPS Publication 198-1, 2008.

[17] J. H. Song, R. Poovendran, J. Lee, and T. Iwata, "*The AES-CMAC Algorithm*," IETF RFC 4493, Jun. 2006.

[18] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA broadcast authentication protocol," *RSA CryptoBytes*, vol. 5, no. 2, pp. 2-13, 2002.

[19] NIST, "*Digital Signature Standard*," FIPS

Publication 186, 1994.

[20] D. Johnson, A. Menezes, and S. Vansto, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Info. Secur.*, pp. 36-63, 2001.

[21] C. Lim and P. Lee, "The korean certificate-based digital signature algorithm," *Comput. & Electr. Eng.*, vol. 25, pp. 249-265, 1999.

[22] Global Positioning Systems Directorate Systems Engineering & Integration, "*NAVSTAR GPS Space Segment/UserSegment L1C Interface*," IS-GPS-800E, Apr. 2018.

[23] J. M. Anderson, et al., "Chips-Message Robust Authentication (Chimera) for GPS Civilian Signals," *ION GNSS+ 2017*, pp. 2388-2416, Sep. 2017.

[24] BeiDou Navigation Satellite System, "*BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B2b v1.0*," Mar. 2020.

[25] Z. Wu, R. Liu, and H. Cao, "ECDSA-Based message authentication scheme for BeiDou-II navigation satellite system," *Trans. Aerospace and Electr. Syst.*, vol. 55, no. 4, Aug. 2019.

[26] Z. Wu, Y. Zhang, and R. Liu, "BD-II NMA&SSI: An scheme of anti-spoofing and open BeiDou II D2 navigation message authentication," *IEEE Access*, vol. 8, 2020.

[27] D. Manandhar and R. Shibasaki, "Authenticating GALILEO open signal using QZSS signal," *ION GNSS+ 2018*, pp. 3995-4003, Miami, Florida, Sep. 2018.

[28] S. Lee, et al., "Prototyping of signal generator for satellite navigation payload," *Electr. Telecommun. Res. Inst. TD*, Jan. 2021.

[29] S. Lee, et al., "Navigation message and early warning service of RNSS," in *Proc. KICS Conf. Commun.*, Jeju Island, Korea, Jun. 2021.

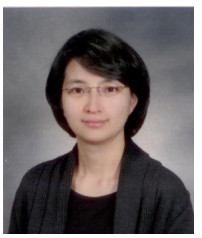
[30] T. Cho, et al., "Authentication for civil navigation message," in *Proc. KICS Conf. Commun.*, Jeju Island, Korea, Jun. 2021.

[31] D. M'Raihi, "*HOTP: An HMAC-Based One-Time Password Algorithm*," IETF RFC 4226, 2005.

조 태 남 (Taenam Cho)

1986년 : 이화여자대학교 전자계산학과 졸업
1988년 : 이화여자대학교 전자계산학과 석사
2004년 : 이화여자대학교 컴퓨터학과 박사
1988년~1997년 : 한국전자통신연구원 위성관제연구실 선임연구원
2004년~2005년 : 이화여자대학교 컴퓨터학과 전임강사
2005년~2017년 : 우석대학교 정보보안학과 교수
2018년~현재 : 우석대학교 IT전자융합공학과 교수
<관심분야> 안드로이드 보안, IoT 보안, 위성보안
[ORCID:0000-0002-5191-0130]

용 승 림 (Seunglim Yong)



1996년 : 이화여자대학교 컴퓨터학과 졸업
1998년 : 이화여자대학교 컴퓨터학과 석사
2006년 : 이화여자대학교 컴퓨터학과 박사
2006년~2007년 : 이화여자대학교 컴퓨터학과 전임강사
2008년~현재 : 인하공업전문대학 컴퓨터시스템학과 교수
<관심분야> 컴퓨터보안, 위성보안
[ORCID:0000-0002-5903-303X]

정 원 찬 (Wonchan Jung)

1986년 12월 : Henderson State University 컴퓨터과학과 졸업
1992년 5월 : Louisiana State University 컴퓨터과학과 박사
1992년 6월~현재 : 한국전자통신연구원 근무
<관심분야> 인공위성 지상국 SW
[ORCID:0000-0001-6740-2643]

이 상 욱 (Sanguk Lee)

1988년 2월 : 연세대학교 천문기상학과 졸업
1991년 3월 : 미 Auburn대학교 항공우주공학 석사
1994년 3월 : Auburn대학교 항공우주공학 박사
1993년 3월~현재 : 한국전자통신연구원, 책임연구원, 현 KPS 위성항법연구센터 센터장

<관심분야> 위성항법, 항공우주공학, 이동통신공학
[ORCID:0000-0002-0744-5032]

유 준 규 (Ryu Joon Gyu)

1999년 2월 : 충남대학교 전파공학과 졸업
2001년 2월 : 충남대학교 전파공학과 석사
2014년 2월 : 충남대학교 전파공학과 박사
2001년 2월~현재 : 한국전자통신연구원 위성광역인프라연구실 실장

<관심분야> 위성통신, 시스템 엔지니어링
[ORCID:0000-0002-1449-4983]