JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 27 No. 12, pp. 209-217, December 2022
https://doi.org/10.9708/jksci.2022.27.12.209

A Study on the Verification of Integrity of Message Structure in Naval Combat Management System

Yong-Gyu Jung*

*Engineer, Naval R&D Center, Hanwha Systems, Gumi, Korea

[Abstract]

Naval CMS(Combat Management System) is linked to various sensors and weapon equipment and use DDS(Data Distribution Service) for efficient data communication between ICU(Interface Control Unit) Node and IPN(Information Processing Node). In order to use DDS, software in the system communicates in an PUB/SUB(Publication/Subscribe) based on DDS topic. If the DDS messages structure in this PUB/SUB method does not match, problems such as incorrect command processing and wrong information delivery occur in sending and receiving application software. To improve this, this paper proposes a DDS message structure integrity verification method. To improve this, this paper proposes a DDS message structure integrity verification method using a hash tree. To verify the applicability of the proposed method to Naval CMS, the message integrity verification rate of the proposed method was measured, and the integrity verification method was applied to CMS and the initialization time of the existing combat management system was compared and the hash tree generation time of the message structures was measured to understand the effect on the operation and development process of CMS. Through this test, It was confirmed that the message structure verification method for system stability proposed in this paper can be applied to the Naval CMS.

▸ **Key words**: Naval Combat Management System, DDS, Hash Algorithm, Hash Tree, Integrity Verification

[요 약]

함정 전투관리체계는 다양한 센서, 무장 장비들이 연동 노드를 통해 연결되며 체계내 노드간 효율적인 통신을 위해 DDS(Data Distribution Service) 통신을 활용한다. DDS를 사용하기 위해 체계내 응용소프트웨어 사이에는 DDS토픽을 기본으로 하는 PUB/SUB(Publication/Subscribe)방식으로 통신한다. 이 PUB/SUB방식으로 통신하는 DDS 메시지 구조가 일치하지 않으면 송수신 응용소프트웨어에서 잘못된 명령처리 및 정보전달 등 문제가 발생한다. 이를 개선하기 위해 본 논문에서는 해시트리를 활용한 DDS 메시지 구조 무결성 검증 방법을 제안한다. 제안하는 방법의 전투관리체계에 적용가능성을 확인하기 위해, 제안하는 방법의 메시지 구조 무결성 검증률을 측정하고 전투관리체계의 운용과 개발과정에 미치는 영향을 확인하기 위해 전투관리체계 초기화 시간 비교, 메시지 해시트리의 생성시간 측정을 하였다. 이 시험을 통해 본 논문에서 제안하는 체계 안정성을 위한 메시지 구조 검증 방법이 함정 전투관리체계에 적용 가능함을 확인하였다.

▸ **주제어**: 함정 전투관리체계, DDS, 해시 알고리즘, 해시 트리, 무결성 검증

• First Author: Yong-Gyu Jung, Corresponding Author: Yong-Gyu Jung
*Yong-Gyu Jung (yonggyu.jeong@hanwha.com), Naval R&D Center, Hanwha Systems
• Received: 2022. 11. 02, Revised: 2022. 12. 14, Accepted: 2022. 12. 21.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

함정 전투관리체계(CMS, Combat Management System)는 전장상황인식을 위해 센서체계로부터 수신한 다량의 정보를 정보처리장치에 전달하여 위협을 식별하고 무장체계에 전술 명령을 전달하여 교전을 수행하고 있다[1]. 함정 전투관리체계의 운용을 위한 메시지의 종류와 양은 센서의 종류, 필요한 체계기능의 수만큼 비례 또는 그 이상으로 증가하며, 대량의 고속 메시지들을 적절한 실시간성을 유지하며 통신을 하기 위해서는 단순한 1:1 네트워크 통신방식보다는 효율적인 통신 미들웨어를 사용하는 것이 필수적이다.

함정 전투관리체계의 응용소프트웨어는 DDS(Data Distribution Service)통신 미들웨어를 활용하여 메시지를 송수신하며 기능을 수행한다. 응용소프트웨어 간 송수신하는 메시지 구조 무결성 위배가 발생하면 통신이 불가하거나 응용소프트웨어의 오류, 운용자가 의도하지 않은 잘못된 명령의 처리가 발생하게 된다. 정확한 표적의 식별과 무장 명령이 생명인 전투관리체계에서 메시지 구조 무결성의 위배는 매우 치명적이라고 볼 수 있기에 정확한 형상관리를 위해 전투관리체계의 신규버전 배포 시 해당 함정의 응용소프트웨어의 전체 메시지 정의 헤더코드를 재생성 및 전체 응용소프트웨어를 재빌드하여 배포 및 설치하여 운용한다. 하지만 연동장비의 메시지 형상변경 또는 함정 전투관리체계의 부분 업그레이드 작업의 경우, 전체 재빌드 작업 및 설치절차는 개발 진행 중인 함정의 한정적인 개발일정, 작전 운용중인 함정의 일정 충돌 등 작업시간의 한계를 겪게 된다.

위 한계를 극복하기 위하여 본 논문에서 부분적인 응용소프트웨어 업그레이드 작업을 수행은 불가피하게 이루어진다. 부분 업그레이드 작업시 개발자 또는 정비자의 실수로 인한 메시지 구조 불일치 소프트웨어 설치 등 전투관리체계의 응용소프트웨어가 안정적인 기능을 수행하기 위한 메시지 구조의 상호 일치 여부를 검증하기 위한 연구를 진행하였다. 본 논문의 구성은 다음과 같다.

2장에서는 관련 연구로써 DDS 기반의 함정 전투관리체계와 무결성 검증에 사용된 해시 알고리즘, 해시 트리 그리고 HEX-BLOOM 모델을 설명한다.

3장에서는 해시 알고리즘과 해시 트리를 메시지 구조 무결성 검증에 적용한 방법에 관해 설명한다.

4장에서는 본 논문에서 제안하는 메시지 구조 무결성 검증률을 측정하고 무결성 검증을 적용한 소프트웨어와 기존 소프트웨어의 초기화 시간 비교 및 검증을 위한 데이터 생성시간을 확인한다.

마지막으로 5장에서 결론으로 논문을 마무리한다.

II. Preliminaries

1. Related works

1.1 Naval Combat Management System With DDS

DDS(Data Distribute Service)통신은 Fig. 1와 같은 PUB/SUB(Publication/Subscribe)형식을 가지는 통신프로토콜이다.

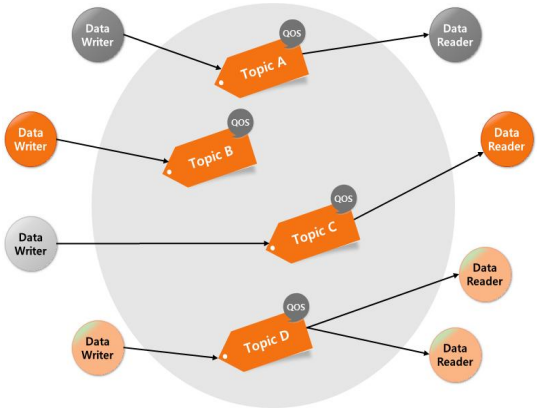

Fig. 1. Publication/Subscribe of DDS Middleware

노드 수가 많은 환경에서 효율적인 메시지 통신을 위해 사용하며, Fig. 2의 구성과 같이 많은 센서와 무장 등 다수의 연동장비를 다루는 연동단이 많은 함정 전투관리체계 내부 노드 간 통신방식으로 적용되고 있다[2-3].

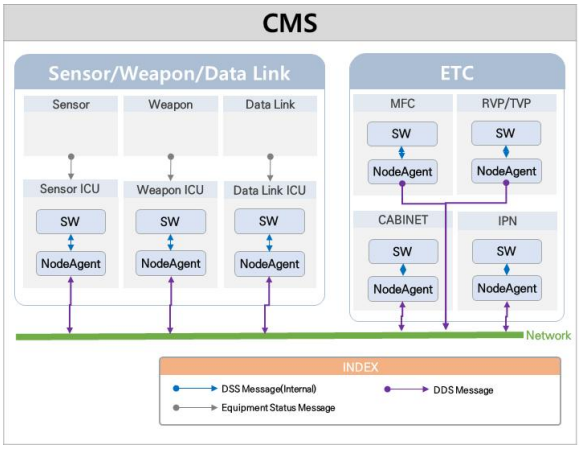

Fig. 2. CMS with DDS

1.2 Hash Algorithm

해시 알고리즘은 네트워크 송수신 메시지의 변조 여부를 확인하기 위해 메시지 내용에서 다이제스트를 도출하는 대표적 알고리즘이며 다음의 두가지 대표적 성질을 만족한다.

1) 일방향성 : 주어진 해시값 h에 대해서 H(x) = h를 만족하는 x를 찾는 것이 계산적으로 불가능

2) 강한 충돌 회피성 : 주어진 x에 대해 H(x) = H(y)를 만족하는 임의의 입력 메시지(y!=x)를 찾는 것이 계산적으로 불가능

일반적으로 널리 쓰이는 해시 알고리즘은 MD5, SHA, HAS-160 등이 있으며 본 연구에서는 암호화 등 보안의 목적보다 상호 메시지 구조의 무결성 검증이 주목적이기 때문에 서로 다른 두 개의 메시지로부터 똑같은 다이제스트를 만들어 내는 것이 계산적으로 불가능한 강한 충돌 회피성을 만족하며, 비교적 연산속도가 빠른 MD5 알고리즘을 활용하였다[4-8].

1.3 Hash Tree

Fig. 3와 같이 리프노드의 해시값으로부터 각각의 부모노드의 해시값을 구하며 상위노드를 구성한 트리를 해시 트리 또는 머클 트리라고 한다.

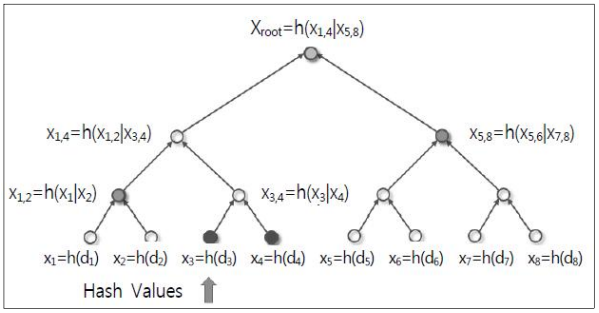

Fig. 3. Hash Tree

리프 노드의 수는 짝수개가 되어야 하며, 마지막 리프노드가 홀수로 끝나는 경우 마지막 노드를 복제하여 부모노드의 해시값을 계산한다.

각 과정을 거쳐서 구한 루트노드는 모든 리프노드의 해시값이 반영된 값으로 볼 수 있으며 리프노드의 값을 일일이 대조하지 않고 루트노드의 값만 일치하여도 모든 리프노드의 값이 일치한다고 볼 수 있다. 이는 데이터의 일치 여부를 확인하기 위해 해시 알고리즘의 강한 충돌회피성을 활용한 하나의 예시이다[7,9].

1.4 HEX-BLOOM

해시 트리 이외 복수의 데이터에 대한 무결성 검증을 위한 모델로 HEX-BLOOM이 있다.

HEX-BLOOM은 집합에 포함되는지 확인하고자 하는 항목에 대해 k가지의 다른 해시 알고리즘을 활용하여 해당 항목이 집합에 포함되는지 확률적으로 확인하는 블룸 필터와 복수의 데이터를 Fig. 4과 같이 XOR연산하여 루트값을 계산하는 LinkedHashX를 활용하여 무결성 검증을 수행한다[10].

1) h_{1} ⊕ h_{2} = h_{12}
2) h_{12} ⊕ h_{3} = h_{123}
3) h_{123} ⊕ h_{4} = h_{1234}
...
4) h_{123...(n-2)(n-1)} ⊕ h_{n} = h_{123...(n-2)(n-1)(n)}
5) LinkedHashX의 루트값 h_{root} = h(h_123...(n-2)(n-1)(n))

Fig. 4. LinkedHashX Root Value Calculation process

HEX-BLOOM은 해시 트리를 운용하기 위해 송수신해야 하는 자료의 크기와 연산속도를 줄여 보완한 모델이지만, 정보의 양을 축약한 만큼 무결성 오류 발생 시 어느 부분의 오류인지 확인이 불가한 단점이 존재한다.

따라서 HEX-BLOOM 모델은 메시지 전체 구조의 무결성 검증에는 활용할 수 있겠지만, 무결성 오류 메시지의 출처 검출이 불가하다.

본 논문에서는 전체 메시지 구조의 무결성 검증과 메시지 구조 무결성 위배 항목 검출 두 가지 조건을 모두 충족하는 해시 트리를 활용하였다.

2. Purpose of Study

2.1 Behavior of Exist CMS Initialization

기존 함정 전투관리체계의 각 노드는 아래와 같은 순서로 초기화 작업을 수행한다.

1. 시간동기화 서버와 시간동기화 완료
2. 응용소프트웨어 DDS 초기화 수행
3. 전투관리체계 응용소프트웨어 간 DDS 메시지 송수신을 통한 기능 수행

각 응용소프트웨어에서 DDS 통신을 위해서 DDS 초기화 작업을 수행하며 PUB/SUB의 형태로 메시지 송수신을 수행하게 된다.

본 연구에서는 “2. 응용소프트웨어 DDS 초기화 수행” 작업 전 메시지 구조의 무결성 검증 절차를 추가하여 초기화 작업을 수행하였다.

2.2 Message Management Tool of DDS Message Struct

메시지 구조 무결성 검증에 필요한 메시지의 해시값 및 해시트리는 함정 전투관리체계 메시지 관리도구에 적용하여 생성하였다.

체계내 응용소프트웨어가 DDS통신을 하기 위해서는 기능설계 단계에서 설계한 메시지 구조를 송수신 시 변환하여 사용한다. 함정에 포함된 기능의 규모마다 다르겠지만, 평균적으로 정의된 메시지의 종류가 약 1000종 이상이며, 전투관리체계 개발자는 효율적인 개발을 위해서 메시지 관리도구를 활용하여 개발하고 있다[11].

메시지 관리도구를 통해 입력된 DDS 메시지는 데이터베이스에 저장되며, 응용소프트웨어 빌드에 필요한 DDS 메시지 구조 헤더파일을 데이터베이스를 기반으로 추출하여 응용소프트웨어 생성에 사용한다. 아래 Fig. 5는 메시지 관리도구를 통해 생성한 DDS 메시지 구조 헤더파일의 구조이다.

```
struct SMsgHeader {
    unsigned long ulMsgID;
    unsigned long ulMsgLength;
    unsigned short unSendCSCI;
    unsigned short unSendCSC;
    unsigned short unSendCSU;
    unsigned short unHWID;
    EDomainID enDomainIndex;
    EModeID enModeInfo;
    unsigned long long llMsgTime;
    unsigned long long llMsgTime_us;
}; //@top-level false
```

Fig. 5. DDS Message Structure Header File

2.3 Expectation Effectiveness

본 논문에서 제시하는 메시지 구조 무결성 검증을 통한 기대 효과는 다음과 같다.

- 메시지 구조 불일치 경고를 통한 체계기능 오류방지
- 응용소프트웨어 통합시험단계에서 개발관리 효율화

메시지 구조 무결성 검증을 통해 개발이나 정비시 잘못된 설치로 인한 메시지 구조 불일치 오류제거를 통한 체계 안정성 확보 그리고 체계통합개발 단계에서 일부 메시지 변경으로 인한 불필요한 전체 빌드 및 설치절차를 줄여 체계 개발단계 효율성 증가를 통한 비용 감소의 효과를 기대한다.

III. The Proposed Scheme

1. Generating Message Structure Hash Tree

전투관리체계 내부 통신에 필요한 메시지 구조 무결성 검증을 위해 메시지 구조의 해시값은 아래와 같의 형태로 생성한다.

<메시지 ID, 메시지 구조 해시값>

메시지 구조 별 해시값 생성은 메시지 관리도구의 메시지 구조 헤더파일 생성 절차를 활용하였으며 생성과정은 Fig. 6과 같다. 생성 코드 중 데이터의 무결성 검증에 필요 없는 문자 제거를 위해 헤더코드 생성 절차에서 띄어쓰기, 구분자(;)의 제외 및 모든 문자열은 대문자로 변경 처리한 값을 입력으로 가지는 해시값을 생성하였다.

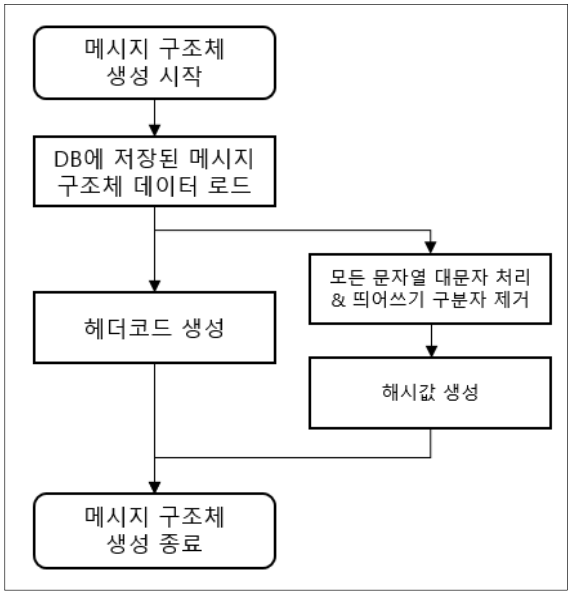

Fig. 6. Hash Values Per Message Generation Process

아래 Fig. 7과 같은 과정으로 생성된 전체 메시지의 해시값을 리프노드로 가지는 메시지 전체에 대한 해시 트리와 각 응용소프트웨어 필수 메시지만 리프노드로 가지는 응용소프트웨어에 대한 해시 트리를 생성한다. 생성된 메시지 전체 대한 해시 트리의 해시 루트값과 각 응용소프트웨어에 대한 해시 트리의 해시 루트값을 헤더파일의 형태로 저장 후 응용소프트웨어 빌드 단계에 포함된다.

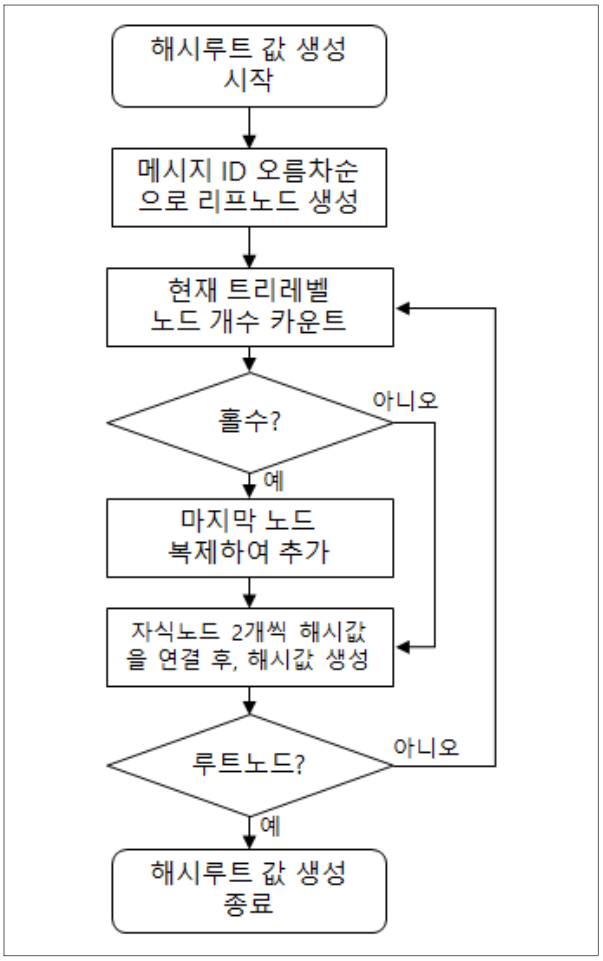

Fig. 7. Hash Root Value Generation Process

각 해시 트리의 갱신은 전투관리체계 메시지 구조 변경 작업 후 메시지 관리도구를 통하여 수정 및 코드 생성 시 변경된 메시지의 해시값을 Fig. 6의 과정을 통해 갱신 후 해당 메시지를 PUB/SUB하는 응용소프트웨어의 해시트리와 전체 해시트리가 헤더파일에 갱신된다.

메시지 변경 작업 후 메시지 구조 해시트리가 자동으로 응용소프트웨어에 반영되도록 통합빌드서버 또는 로컬환경에서 응용소프트웨어 빌드를 위한 DDS 메시지 구조 헤더파일 생성 시 해시트리를 생성하도록 설계하였다.

2. Message Structure Integrity Verification Procedure

메시지 구조 무결성 검증 수행을 위해서 검증 절차를 수행할 검증 노드가 필요하다. 때문에 함정 전투관리체계에서 시간동기화를 위해 항상 활성화 되어있는 OSD(Own Ship Data, 함기준센서)노드(이하 검증 노드)를 검증 노드로 활용하였다. 메시지 구조 검증 노드는 Fig. 8과 같이 구성하였으며 UDP 멀티캐스트 통신을 통해 메시지 구조 무결성 검증을 수행한다.

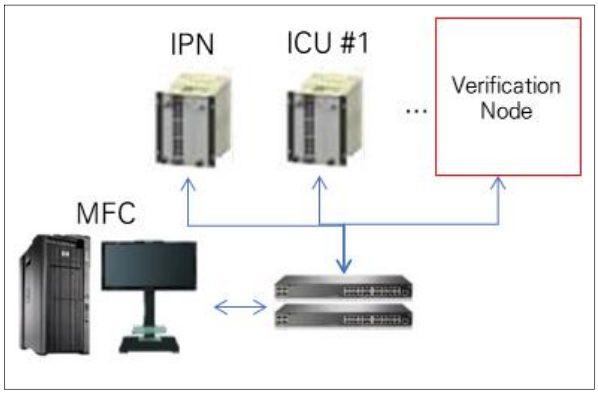

Fig. 8. Verification Node

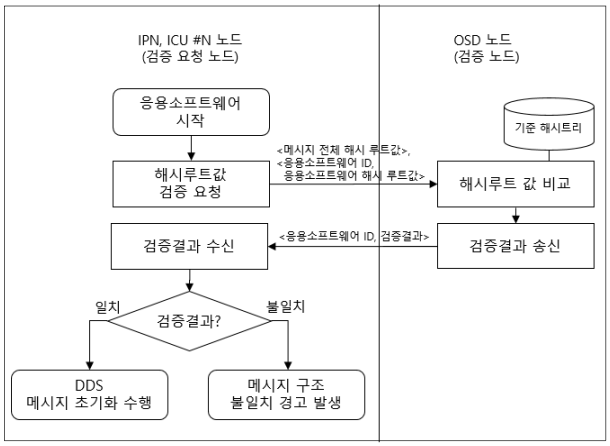

Fig. 9. Message Structure Integrity Verification Procedure

DDS 메시지 초기화 전 메시지 구조 무결성 검증을 위해서 Fig. 9의 절차를 수행한다. 메시지 구조 무결성 검증을 요청하는 노드에서 무결성 검증을 수행해줄 검증 노드로 <응용소프트웨어 ID, 메시지 전체 해시 루트값, 응용소프트웨어 해시 루트값>을 송신한다. 메시지 전체 해시 루트값이 일치하는 경우 또는 응용소프트웨어 해시 루트값이 일치하는 경우 메시지 구조 일치 응답을 검증요청 노드로 보낸다. 두가지 해시 루트값이 모두 불일치하는 경우 메시지 구조 불일치 응답을 검증요청 노드로 보낸다.

메시지 구조가 일치하는 경우 전투관리체계 운용을 위한 DDS 메시지 초기화 작업을 수행하고, 메시지 구조가 불일치 하는 경우 전투관리체계의 오류를 발생 시킬 수 있으므로 초기화 작업을 수행하지 않고 MFC(Multi Function Console, 다기능콘솔)의 경고팝업과 체계 소프트웨어 감시를 통하여 운용자에게 오류를 알려준다.

작전 운용중인 함정에서 해시루트값 불일치의 경우 정비 또는 부분 소프트웨어 업그레이드시 나타날 수 있다. 이는 전투관리체계 응용소프트웨어 형상 불일치로 인해 메시지 구조의 불일치 외 기능적 오류의 가능성을 경고를

통해 사용자에게 알려주고 적절한 정비를 받을 수 있게 도와준다. 더불어 메시지 불일치로 인한 검증정보는 파일의 형태로 저장하여 함정 전투관리체계의 형상 현황관리에 활용한다.

실제 운용중인 환경과 달리 개발단계에서는 좀 더 효율적인 개발 및 기능검증의 관점으로 앞서 메시지 구조 검증 과정 외 추가적인 기능으로 Fig. 10의 절차와 같이 개별 메시지 구조 무결성 검증기능을 활용한다.

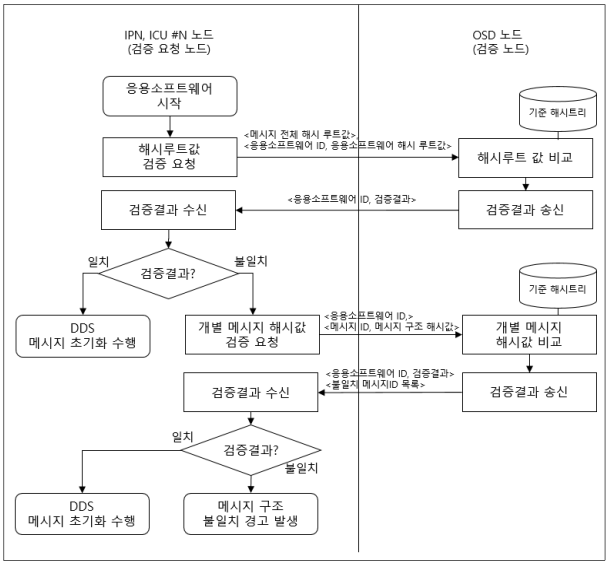

Fig. 10. Message Structure Integrity Verification Procedure for Development

전체 메시지 구조의 형상이 일치하지 않더라도 응용소프트웨어에서 사용되는 메시지의 구조가 검증 노드와 일치하는 경우 메시지 구조에 의한 오류는 발생하지 않기 때문에 전투관리체계 개발단계에서 해당 기능을 활용하여 메시지 변경에 따른 부분 패치 및 체계 기능시험에 활용할 수 있다.

개별 메시지 무결성 검증은 응용소프트웨어에서 사용되는 메시지의 해시값을 <메시지 ID, 메시지 구조 해시값>의 형태로 검증 요청한다. 검증 노드에서 각 메시지의 해시값이 일치하는지 검증하며 모두 일치 또는 불일치와 불일치 메시지 목록을 응답한다. 불일치 메시지 목록은 개발자가 불일치하는 메시지를 식별하여 부분적인 메시지 구조 갱신에 활용한다.

전투관리체계 시험개발환경에서는 실제 함정의 구성과 가능한 대부분 일치하도록 구성하여 체계 응용소프트웨어들이 실제 함정에서의 작전 시나리오대로 운용될 때 체계 내 기능들이 정확하게 기능하는지 확인한다. 개발 및 시험평가 단계에서는 DDS 메시지 구조 변경이 다소 빈번하게 발생하며 체계 응용소프트웨어 간 메시지 구조 불일치의 가능성이 존재한다.

메시지 구조 불일치를 해소하기 위해서 체계 전체 메시지 헤더 재생성 및 체계 전체 응용소프트웨어의 재빌드 및 설치 또는 변경된 메시지 구조를 사용하는 체계 응용소프트웨어의 식별, 개별 메시지 헤더 재생성, 재빌드 및 패치를 거쳐야 체계 응용소프트웨어 간 메시지 구조 무결성을 보장할 수 있게 된다. 위와 같은 메시지 구조 불일치 해소를 위한 활동은 체계의 규모에 비례하여 시간과 노력이 많이 들어가게 되는 활동이기 때문에 효율화가 필요한 부분이다. 응용소프트웨어 메시지 구조의 무결성을 보장할 수 있게 되면 메시지 구조 변경에 의한 전체 응용소프트웨어 재생성 및 설치의 활동을 줄일 수 있어 전투관리체계 개발관리를 효율화할 수 있다.

메시지 구조 검증 요청 및 응답 메시지 구조는 Fig. 11과 같다. 각 메시지는 메시지의 종류, 송수신 노드, 메시지 길이, 송신시간을 기록한 SMSG_HEADER 구조체를 포함하며, 각 필드는 아래와 같이 구성하였다.

1. 메시지 구조 검증 요청 메시지
   - SW_ROOT_HASH_VERIFY_REQUEST
   1) 응용소프트웨어 ID(ulSWID)
   2) 메시지 전체 해시루트값(ulSWRootHash[4])
   3) 응용소프트웨어 해시 루트값(ulAllRootHash[4])

2. 메시지 구조 검증 응답 메시지
   - SW_ROOT_HASH_VERIFY_ACK
   1) 응용소프트웨어 ID(ulSWID)
   2) 구조 검증 결과(ulVerifyResult)

3. 개별 메시지 구조 검증 요청 메시지
   - SW_MSG_HASH_VERIFY_REQUEST
   1) 응용소프트웨어 ID(ulSWID)
   2) 메시지 개수(ulMsgCount)
   3) 개별 메시지 해시값(ulMsgIDHash[1000][5])

4. 개별 메시지 구조 검증 응답 메시지
   - SW_MSG_HASH_VERIFY_ACK
   1) 응용소프트웨어 ID(ulSWID)
   2) 구조 검증 결과(ulVerifyResult)
   3) 메시지 개수(ulMsgCount)
   4) 불일치 메시지ID 목록(ulDeserializeMsgID[1000])

```
struct SW_ROOT_HASH_VERIFY_REQUEST
{
    SMSG_HEADER             stMessageHead;
    unsigned long           ulSWID;
    unsigned long           ulSWRootHash[4];
    unsigned long           ulAllRootHash[4];
};

struct SW_ROOT_HASH_VERIFY_ACK
{
    SMSG_HEADER             stMessageHead;
    unsigned long           ulSWID;
    unsigned long           ulVerifyResult
};

struct SW_MSG_HASH_VERIFY_REQUEST
{
    SMSG_HEADER             stMessageHead;
    unsigned long           ulSWID;
    unsigned long           ulMsgCount;
    unsigned long           ulMsgIDHash[1000][5];
};

struct SW_MSG_HASH_VERIFY_ACK
{
    SMSG_HEADER             stMessageHead;
    unsigned long           ulSWID;
    unsigned long           ulVerifyResult
    unsigned long           ulMsgCount;
    unsigned long           ulDeserializeMsgID[1000];
};
```

Fig. 11. Verification Request and Response Message Structure

IV. Software Evaluation

본 논문에서 제안한 메시지 무결성 검증방법이 전투관리체계에 적용 가능함을 확인하기 위해 다음과 같이 시험을 수행하였다.

메시지 구조 무결성 검증기능 확인을 위해 메시지 구조를 임의로 변경한 응용소프트웨어의 무결성 검증률을 측정하였다. 검증기능의 전투관리체계에 영향을 측정하기 위해 검증절차를 추가한 전투관리체계와 기존 전투관리체계의 초기화 시간 비교 및 검증에 필요한 메시지별 해시값, 해시트리 생성시간을 확인하였다. 평가 실험에 대한 환경은 아래 Table 1, Fig. 12와 같다.

Table 1. Test Environment

Item	MFC	IPN	ICU
CPU	Intel Core i7-6700 @3.40GHz	Intel Core i7-4700 EQ @2.40GHz	Intel Core i7 E610 @2.53GHz
Memory	8GB	8GB	4GB
OS	Window 7	RTST Linux	RTST Linux

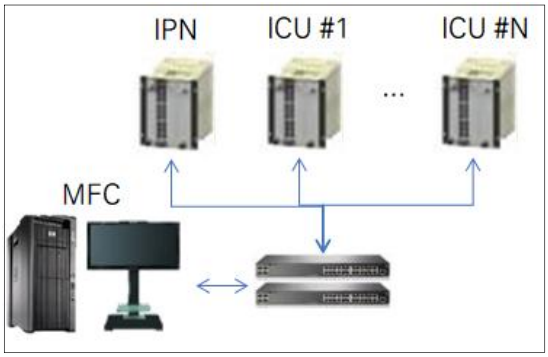

Fig. 12. Test Environment

1. Message Structure Verification Rate Measurement

임의의 응용소프트웨어 메시지 구조 변경 후 메시지 구조 무결성 검증기능을 확인하였다. Table 2는 메시지 구조 무결성 검증률을 측정한 결과이다.

실험에 사용한 응용소프트웨어는 32종(총 DDS 메시지 514종류)을 활용하였으며 32종의 응용소프트웨어를 아래의 3가지 시나리오로 각 50회, 총 150회의 메시지 구조 변경작업 및 무결성 검증을 시험하였다.

1) 임의의 메시지에 필드의 순서를 변경한 경우
2) 임의의 메시지에 필드의 자료형을 변경한 경우
3) 임의의 메시지에 새로운 필드를 추가한 경우

Table 2. Message Structure Verification Rate

Test Case	Verification Rate
case 1) Changing Field Order	100%
case 2) Changing Field Type	100%
case 3) Add New Fields	100%

메시지 구조 변경작업을 수행한 모든 응용소프트웨어를 무결성 검증기능을 통해 식별할 수 있었다. 이를 통해 제안된 방법이 메시지 구조 무결성 검증이 정확하게 수행됨을 확인할 수 있었다.

2. CMS Initializing Time after Applying Integrity Verification

해시 트리를 활용한 무결성 검증절차를 수행했을때 전투관리체계 응용소프트웨어에 영향을 측정하기 위해 전투관리체계 초기화를 완료하는데 걸리는 시간을 기존 응용소프트웨어와 비교하였다.

응용소프트웨어의 초기화 시간은 DDS 초기화 완료 및

기능 수행을 위한 초기 데이터 송수신을 모두 완료한 시간을 측정하였다. Table 3은 기존 응용소프트웨어와 검증절차를 수행한 응용소프트웨어의 초기화 완료 시간을 비교한 결과이다.

실험에 사용한 응용소프트웨어는 32종(총 DDS 메시지 514종류)을 활용하였으며 아래 2가지의 경우를 50회 수행한 평균을 계산하였다.

1) 전체 메시지 해시 루트값 또는 응용소프트웨어의 해시 루트값이 일치하는 경우,
2) 전체 메시지 해시 루트값, 응용소프트웨어의 해시 루트값 모두 불일치 하나 메시지 각각의 해시값은 일치하는 경우

Table 3. Compare time to CMS Initialization

Test Case	Exist Software	Proposed Software
case 1) Message Initialization Time	10.23 sec.	10.30 sec.
case 2) Message Initialization Time		10.37 sec.

전투관리체계 초기화 처리 시 메시지 구조 무결성 검증 기능을 추가하였음에도 98%이상의 전투관리체계 초기화 처리성능을 확인하였다.

해당 작업은 전투관리체계 초기 구동 시에만 발생하는 차이이며 함정의 전투관리체계 작전운용 관점에서 함정의 출동 및 작전 준비의 시간을 고려해보았을 때 위 성능은 운용에 적합한 성능이라 볼 수 있다.

이를 통해 메시지 구조 무결성 검증기능을 함정 전투관리체계에 적용 가능함을 확인하였다.

3. Time Required to Generate Data for Integrity Verification

메시지 구조 무결성 검증을 위한 데이터 생성시간에 대한 실험은 MFC(Multi Function Console, 다기능콘솔) 환경에서 수행하였다.

Table 4는 무결성 검증을 위한 메시지 개수별 데이터 생성에 걸리는 시간을 실험한 결과이다. 데이터 생성 시간 측정은 992개의 메시지를 각 총 50회 수행한 평균을 계산하였다.

Table 4. Compare Time of Hash Tree Generation

-	Exist Software	Proposed Software
Time of Generation(sec.)	3.73 sec.	4.11 sec.

992개의 리프노드를 가지는 해시트리 생성 시 1985번의 MD5 해시 연산을 수행하였고, 코드생성 시간은 기존 대비 약 110%정도 소요되었다. 해당 작업을 고성능의 전투관리체계 빌드 서버 환경에서 수행 시 파일생성 성능의 차이는 더욱 줄어들 것으로 예상된다.

이를 통해 메시지 구조 무결성 검증을 위한 해시 트리 생성절차를 함정 전투관리체계 메시지 관리도구에 적용 가능함을 확인하였다.

V. Conclusions

전투관리체계는 메시지 송수신을 통해 체계기능을 수행하므로 응용소프트웨어간 메시지 구조의 무결성은 매우 중요하다. 전투관리체계의 기능오류는 작전실패로 이어질 수 있는 만큼 체계 안정성의 관점에서 메시지 구조는 정확하게 일치하여야 한다.

현재 소프트웨어의 안정적인 기능 수행을 위해 많은 프로세스 도입 및 노력을 기울여 운용하고 있지만, 정비의 실수 또는 예측하지 못한 여러 가지 휴먼에러 등의 이유로 메시지 구조의 불일치는 발생할 수 있다.

본 논문에서는 전투관리체계의 메시지 구조 불일치로 인한 문제점을 해결하기 위해 메시지 구조 무결성 검증방안 설계, 구현, 무결성 검증기능 확인, 무결성 검증 적용에 따른 영향분석을 확인하여 체계 안정성을 위한 메시지 구조 무결성 검증방법의 전투관리체계 적용 가능성을 확인하였다.

추후 연구로는 전투관리체계의 기능 운용을 위한 메시지 트랜잭션/시퀀스의 문맥 무결성 검증방법에 대한 연구를 할 예정이다.

REFERENCES

[1] Ko, Soon Joo, Park, Do Hyun. "An Examination on Overseas Technology Trend and Domestic Development Pattern of the Naval Combat Management System." Journal of the Korean Association of Defense Industry Studies, Vol. 16, No. 2, pp.237-258, Dec. 2009.

[2] Kitae Kim, Seunghyun Yang, Wonwoo Jung. “Case Study of Real-time Data Sharing Framework Based on Data Distribution Service Middleware”, KIISE Transactions on Computing Practices 26(4), 202-210(9pages), Apr. 2020, DOI : 10.5626/KTCP.2020.26.4.202

[3] Ji-Yoon Park, Moon-Seok Yang, Dong-Hyeong Lee. "A Study on IISS Software Architecture of Combat Management System for improving modifiability" Journal of The Korea Society of Computer and Information, Vol. 25 No. 2, pp. 133-140(8pages), Feb. 2020.

[4] Federal Information Processing Standards Publication (FIPS PUB), "Secure Hash Standard", Vol 57, No.21, pp.3747-3749, Jan. 1992.

[5] M. J. B. Robshaw, "MD2, MD4, MD5, SHA and Other Hash Function", Technical Report TR-101, Version 3.0, RSA Laboratories, Jul. 1994.

[6] PKCS #1 v2.1 "RSA Cryptography"

[7] Hyun-Kyoung Yoon, Wan-Kyung Kim, Woo-Young Soh. “Design and Implementation of Reliable Web Authentication System Using MD5 and Crypt”, Korea Multimedia Society, pp.87-90(4pages), Daejun, South Korea, May. 2004.

[8] Ju-Dae Hyun and Byeong-Yoon Choi. “Design of Processor for MD5 Hash Algorithm” pp.1,893-1,896(4pages), South Korea, Jul. 2002.

[9] Seung Kyu Park, Jong Jin Park, and Dong Wook Lee. “Implementation of the Large-scale Data Signature System Using Hash”, Journal of The Institute of Electronics and Information Engineers Vol.55, NO.5, pp.43-50(8pages), May 2018. DOI : 10.5573/ieie.2018.55.5.43

[10] Ripon Patgiri and Malaya Dutta Borah, “HEX-BLOOM: An Alternative to the Merkle Tree”, Cryptology ePrint Archive: Report 2021/773, 2021, https://ia.cr/2021/773

[11] Juwon Lee, "Development of Message Define & Management System based on Distributed Processing Environment for Naval Combat Systems" KIISE Transactions on Computing Practices, Vol. 23, No. 12, pp. 670-676, Dec. 2017. DOI : 10.5626/KTCP. 2017.23.12.670

Authors

Yong-Gyu Jung received the B.S. degrees in Computer Engineering from Yeungnam University, Korea, in 2014. He is currently working in Hanwha Systems Co. from 2014. He is interested in Naval Combat System, Tactical Data Link System and so on.