JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 29 No. 12, pp. 189-197, December 2024
https://doi.org/10.9708/jksci.2024.29.12.189

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

Young Choi*

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

[Abstract]

The Naval Combat Management System (CMS) is an automated system that networks all sensors and armaments on a ship to efficiently respond to threats. Data transfer within the system is performed through the Data Sharing Service (DSS), and errors in this process can severely affect system performance, making accurate validation essential. This study proposes a new verification method to store message transaction data between devices in the system and verify their integrity in actual operation. We utilize Merkle tree, a data structure that uses Hash Functions, to verify data matching, and experimentally measure the verification performance and system impact. The results confirm that the message transaction verification method proposed in this paper for verification efficiency and system stability is applicable to Navel CMS.

▸ **Key words**: Naval Combat Management System, DSS, Hash Function, Merkle Tree, Integrity Verification

[요 약]

함정 전투관리체계(Combat Management System/CMS)는 함정에 있는 모든 센서와 무장을 네트워크로 연결하여 효율적으로 위협에 대응하는 자동화 시스템이다. 체계 내의 데이터 전달은 DSS(Data Sharing Service/DSS)를 통해 수행하고 있으며 이 과정에서의 오류는 시스템 성능에 심각한 영향을 미칠 수 있어 정확한 검증이 필수적이다. 본 연구는 체계 내 장치들 간의 메시지 트랜잭션 데이터를 저장하고, 실제 운용 시 무결성을 확인하는 새로운 검증 방법을 제안한다. 해시함수를 사용하는 데이터 구조인 머클 트리를 활용하여 데이터 일치 여부를 확인하며, 검증 성능과 시스템 영향력을 실험적으로 측정하였다. 이 시험을 통해 본 논문에서 제안하는 검증 효율성과 체계 안정성을 위한 메시지 트랜잭션 검증 방법이 함정 전투관리체계에 적용 가능함을 확인하였다.

▸ **주제어**: 함정 전투관리체계, DSS, 해시 함수, 머클 트리, 무결성 검증

• First Author: Young Choi, Corresponding Author: Young Choi
*Young Choi (young0515@hanwha.com), Naval R&D Center, Hanwha Systems
• Received: 2024. 11. 08, Revised: 2024. 11. 25, Accepted: 2024. 11. 28.

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

I. Introduction

함정 전투 관리체계(Combat Management System/ CMS)는 함정에 탑재되는 모든 센서, 무장, 항해 지원 장비 등을 네트워크로 연결하여 각각의 독립적인 기능과 성능 및 특성을 유지하면서도 통합된 무기체계로서 다양한 적의 위협에 대해 정확한 탐지 및 대응 능력과 동시 다발적인 전투 상황 하에서 함 탑재 센서와 외부로부터 획득된 정보를 종합 처리하여 최적의 전투 능력을 제공할 수 있도록 표적의 탐지, 추적에서부터 위협평가, 무장 할당, 교전 및 명중 여부 평가 분석에 이르기까지 지휘 및 무장통제 기능이 통합된 자동화 전투체계이다.[1-2]

네트워크로 연결된 각각의 장비들은 그에 대응되는 연동 처리장치와 전시 처리장치를 통해 운용자에게 그 임무를 수행 가능토록 구성되어 있다. 운용자는 전시 처리장치에 전시된 입력 컨트롤을 사용하여 담당 장비에 대한 운용을 수행하며 운용 상 필요한 정보들 역시 전시 컨트롤에 전시된 정보를 바탕으로 취합하여 임무를 수행 한다. 이 과정에서 연동대상 장비와 연동 처리장치, 전시처리장치 간의 데이터는 TCP/IP, UDP/IP, Serial, DDS 등의 통신 프로토콜을 통해 유통된다.

전투체계는 네트워크를 통하여 연결되면서 운용자의 임무를 수행하고 있다. 운용자의 직접적인 육안 통제를 통해 임무를 수행하는 하드웨어 장비에 비해서 전투체계는 운용자의 제어를 받는 전시화면이 지휘통제실 내에 있으면서 실제로 명령에 의해 동작을 수행하는 연동 장비는 외부에 있는 체계이다. 그렇기 때문에 전투체계는 외부에 있는 연동 장비의 정확한 동작을 위해 오류, 기능 미작동, 의도하지 않은 명령 처리가 수행되지 않도록 동작하여야 한다. 하지만 체계 개발 중 개발자의 휴먼 에러, 불명확한 설계 사항 등의 이유로 소프트웨어에 의한 오류 및 오작동 등의 문제는 필연적으로 발생한다. 이러한 오작동을 유발하는 원인 중 하나로는 네트워크 상의 부정확한 메시지 전달이 있으며, 체계 개발자들은 전투체계에 부정확한 메시지를 전달하는 오류가 있는지 식별하기 위해 매 개발단계에서 검증 시험을 수행하고 있다.

네트워크 상의 메시지 전달에 문제가 발생하여 SW 문제가 발생한다면 엔지니어는 해당 문제의 원인을 식별하기 위해 데이터 로그 탐색이나 운용자의 현상 재현을 통해 정확히 어떤 전투체계 메시지 처리 과정에서 오류가 식별되었는지 확인이 필요하다. 하지만 하나의 연동 장비에 대해서도 다수의 메시지 상호작용이 발생하는 전투체계 시스템 상에서 메시지를 식별하기는 간단할 수 있지만 해당 메시지 안의 필드 단위의 트랜잭션에 문제가 발생한다면 문제의 원인을 파악하는데 많은 작업 공수와 개발 비용이 소요된다.

위 상황을 개선하기 위해 본 문에서는 메시지 트랜잭션 무결성 오류로 인하여 나타나는 SW 결함에 대한 원인 파악을 신속하게 할 수 있도록 하는 방법을 제시한다. 제시한 방법은 전시 처리장치와 연동 처리장치 단계의 정상적인 메시지 트랜잭션을 저장하여, 추후 장비 운용 시의 트랜잭션 오류가 발생한다면 저장된 트랜잭션과의 상호일치 여부를 판단하여 검증한다. 구성은 다음과 같다.

2장에서는 관련 연구로써 전투체계 개발 및 네트워크 트랜잭션 검증에 대한 기술적 배경을 제시한다.

3장에서는 머클 트리를 활용한 메시지 트랜잭션 무결성 검증을 전투체계에 적용 및 통합한 과정을 제시한다.

4장에서는 시험 데이터를 검증 데이터와 비교하여 불일치를 검출하는 관점에서 제안한 무결성 검증 방안의 검증률을 측정하고 기존 소프트웨어 운용 상에서 무결성 검증이 동시 수행되는 동안의 퍼포먼스 영향력을 확인한다.

마지막 5장에서는 결론 및 추후 과제로 본 논문을 마무리하였다.

II. Preliminaries

1. Background

1.1 Naval Combat Management System

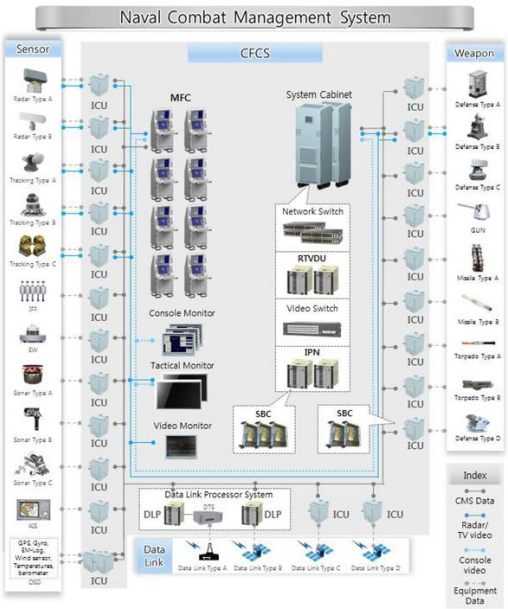

Fig. 1. Naval CMS architecture

함정 전투체계는 Fig. 1과 같이 인간의 두뇌처럼 레이더, 소나 등의 센서와 함포, 유도탄 등의 무장을 통합하여 작전에 필요한 정보를 자동으로 수집하고 전술 상황 평가, 지휘 결심, 위협 평가, 무기 할당, 교전 등의 기능을 효율적으로 수행하는 무기체계이다.

지휘무장통제체계(Combat Fire Command System/CFCS), 센서, 무장, 데이터링크로 구성되며, 이 가운데 핵심적인 역할을 하는 CFCS는 정보 처리장치, 전시 처리장치, 연동 단위, 통합 네트워크, 영상 분배 장치 등으로 구성되어 있다. 생존성과 가용성을 높이기 위해 다수의 정보처리장치와 연동 처리장치(Iterface Control Unit/ICU)들과 운용자의 명령 처리를 위한 전시 처리장치가 네트워크적으로 연결되어 있다.[3]

1.2 DSS(Data Sharing Service)

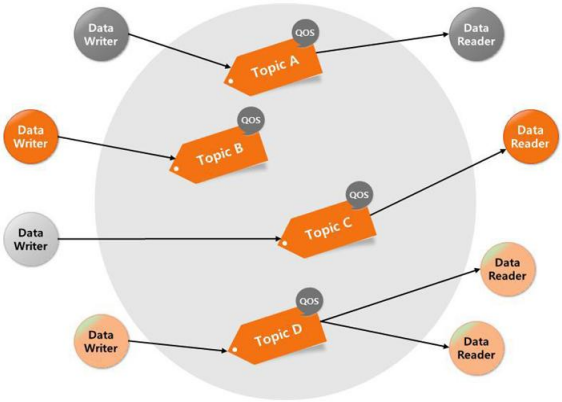

Fig. 2. DDS Middleware

DSS(Data Sharing Service/DSS) 통신은 Fig. 2와 같이 구성된 미들웨어인 DDS(Data Distribution Service/DDS)를 기반으로 만든 통신 서비스이다. 함정 전투체계는 여러 연동 장치들이 DSS를 통해 네트워크 적으로 연결되어 있는데 이 과정에서 서로 주고받기 위한 메시지를 주제(Topic)로 정의하고 작성자(DataWriter)를 통해 메시지를 작성하고 독자(DataReader)를 통해 작성된 메시지를 확인한다.. 각 연동장치는 이렇게 작성된 토픽을 발행 및 구독(Publisher/Subscriber) 방식으로 주고받으면서 동작한다.[4]

이처럼 함정 전투체계는 여러 구성 장치 간에 주고 받는 DSS 데이터를 사전 정의된 필드들로 구성된 구조체를 기반으로 운용 된다.[5] 복합적인 전투체계 구조 상에서 주고받는 필드들은 각 장비들의 동작을 수행하는 기반 데이터이기 때문에 주고 받는 과정이 잘못되어진다면 운용자가 의도하지 않은 동작이 발생할 수 있고, 연동 처리장치의 비정상적 종료와 같은 치명적인 상황도 발생할 수 있다.

예를 들어 함포 교전에 대하여 표적에 대한 잘못된 데이터가 처리되어 급박한 상황에 함포가 불발될 수 있으며, 선박 보조 장치 등에 의해 수신된 운동 정보 처리에 오류가 발생하여 함 운항이 정상적으로 수행되지 않을 수 있다. 전투체계는 이러한 운용 상 치명적인 상황에 대하여 기본적으로 대체 방안도 수립되어 있고 수많은 예외 처리도 반영되어 있지만 기본적으로 예외 자체가 발생하지 않는 것이 체계 안정성에 도움이 될 수 있으므로 본 논문에서는 해시 함수를 활용한 데이터 구조 중 하나인 머클 트리를 통해 DSS 데이터 필드 단위의 해시화를 통해 전투체계 메시지 트랜잭션 무결성을 검증하고자 한다.

1.3 Hash Function

해시 함수는 임의의 길이를 가진 문자열을 입력으로 받아 해시 값 또는 다이제스트라고 하는 문자열을 출력하는 함수이다. 해시 함수는 보안 속성을 만족하도록 설계되었으며, 다음과 같은 중요한 속성들이 있다.

1) 제 1 역상 저항성 : 주어진 해시 값 h에 대해, $H(M)=h$를 만족하는 원래의 입력값 M을 찾는 것이 계산적으로 불가능하다.

2) 제 2 역상 저항성 : 주어진 값 M과 $H(M)$에 대해서 $H(M) = H(M')$을 만족하는 M' 값을 찾는 것이 계산적으로 불가능하다.

3) 충돌 저항성 : 임의의 두 입력 M과 $M'(M \neq M')$이 $H(M) = H(M')$를 만족하는 M과 M'를 찾는 것이 계산적으로 불가능하다.

제 1 역상 저항성에 의해 해시값을 통해 원본의 문자열을 계산해내는 것이 불가능하며 제 2 역상 저항성을 통해 동일한 해시 값을 가지는 또 다른 입력값을 계산해내는 것이 불가능하고 충돌 저항성에 의해 서로 다른 두 문자열이 같은 해시값을 가질 수 없으므로 동일한 해시값으로 인한 데이터 중복이 발생할 수 없으며 해시값을 역추적하여 원래의 데이터를 유추할 수 없다고 볼 수 있어 전투체계 데이터 필드를 해시 함수를 통해 암호화하여 처리하고 데이터 구조를 활용하여 검증을 수행하고자 하였다.[6]

1.4 Integrity Verification

해시 함수를 활용하여 데이터의 일치 여부를 확인하는 데이터 구조로는 대표적으로 블룸 필터와 머클 트리가 있다.

블룸 필터는 데이터의 집합에서 특정 원소의 존재 여부를 효율적으로 검사할 수 있는 확률적 데이터 구조이다.

고정된 크기의 비트 배열과 n개의 해시 함수를 사용하여 원소를 저장하기 때문에 공간 효율성이 뛰어나고 빠른 연산이 가능하다. 블룸 필터는 내부적으로 존재하는 원소에 대해서 실제로 존재하는 원소의 부재 여부 상태는 정확하게 판단이 가능하지만 존재하지 않는 원소의 존재 여부 상태에 대해서는 오판이 발생할 수 있다.[7] 즉 시험 데이터와 검증 데이터가 주어졌을 때 두 데이터가 일치하여 일치성에 대한 검증을 빠르고 정확하게 판별해낼 수 있지만 검증 데이터와 불일치한 시험 데이터에 대해서도 일치하다고 판단될 수 있다. 이러한 특징 때문에 전투체계 운용에 치명적일 수 있는 메시지 트랜잭션의 오류를 식별하기에 적합하지 않다.

머클 트리는 데이터 집합의 무결성을 효율적으로 검증하기 위해 사용되는 트리 형태의 데이터 구조이다. 데이터들을 작은 청크로 분할하고 분할된 각 청크를 해시화하여 트리 구조의 노드로 저장한다. 머클 트리의 노드는 실제 데이터 청크를 해시화한 리프 노드(Leaf Node), 리프 노드를 자식으로 가지며 각 리프 노드의 해시값을 결합하여 다시 해시화하여 저장하는 비리프 노드(Non-Leaf Node), 최종적으로 이러한 과정을 반복하여 만들어진 최상위 비리프 노드인 머클 루트(Merkle Root)로 구성된다.[8]

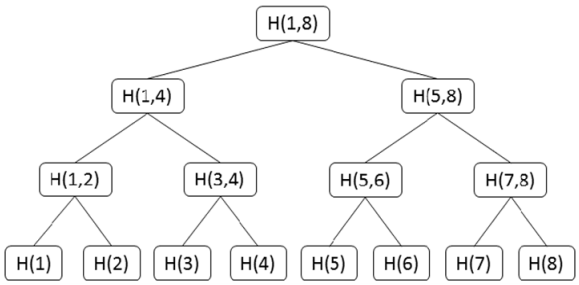

Fig. 3. Merkle Hash Tree

Fig. 3은 머클 트리의 기본적인 구조이다.[9] 최하위의 노드는 리프 노드이며 그 위를 각 리프 노드를 결합한 후 다시 해시화한 비리프 노드가 위치하고 있고, 최상위 노드는 머클 루트이다. 머클 트리는 기본적으로 이진 트리의 구조를 가지기 때문에 리프 노드의 개수는 일반적으로 짝수개이다. 리프 노드의 개수가 홀수이면 마지막 노드 복사, 임의의 패딩(Padding) 값을 사용하여 노드의 개수를 짝수로 유지한다.

머클 트리의 노드는 기본적으로 모두 해시값이므로 충돌 저항성에 의해 각 노드들이 동일한 해시값을 가질 수 없으므로 각 노드의 집합인 머클 루트는 기본적으로 유일한 값으로 볼 수 있으며 데이터의 변조가 발생했다면 자연적으로 머클 루트의 값도 변경되기에 데이터의 변조를 탐지하기 용이하다.

본 논문에서는 이러한 머클 트리의 특성을 이용하여 시험 데이터와 검증 데이터를 머클 트리화하여 머클 루트를 통해 데이터의 차이 여부를 검증하고 리프 노드 단위의 비교를 통해 변질된 데이터를 검출하였다.

2. Related Works

함정 전투체계 내의 소프트웨어 무결성에 대한 연구는 수차례 진행 되었지만 데이터 필드 단위의 무결성 검증에 대한 설계를 제시한 사례는 없었다.

2.1 Enhancing Installation Security for CMS

해시 함수를 통한 소프트웨어 형상에 대한 무결성 검증에 대한 연구[10]가 진행되었다. 해당 연구는 각 함정에 설치하는 전투체계 소프트웨어에 대한 전체적인 형상 무결성에 대한 검증을 수행하여 그 성능을 만족시켰으나, 소프트웨어의 임무 수행 과정에서의 무결성을 보장하지 않는다.

2.2 Integrity of Message Structure in CMS

해시 함수와 머클 트리를 이용한 전투체계 메시지 구조 무결성 검증에 대한 연구[11]가 진행되었으나, 해당 연구는 전투체계의 구조 자체에 대한 무결성 검증이였으며, 이는 소프트웨어가 실제 운용되는 상황이 아닌 초기 실행 단계에서의 연동 장비와 주고받기 위한 데이터 구조에 대한 검증 설계이므로 데이터 교환에 대한 무결성을 검증할 수 없다.

이처럼 해시 함수와 이를 활용한 데이터 구조를 이용하여 전투체계 내의 무결성을 확보하는 방안은 여러 가지가 있지만, 대부분의 연구는 소프트웨어 실행 초기 단계에서의 무결성 검증이 다수를 차지하고 있다. 이는 소프트웨어의 운용 시작 단계에서의 오류를 검출하기에 용이하지만 전투체계 소프트웨어와 연동장치 간의 상호 작용에 발생하는 오류를 검출하기에 한계가 있다.

3. Purpose of Study

3.1 Message Transaction in CMS Software

함정 전투체계는 체계 내에 센서, 무장, 데이터링크와 같은 연동 대상 장비가 탑재되고 장비와 연동하여 정보를 처리하는 연동 처리장치를 구성하여 장비를 제어하고 운용자가 존재하는 장비의 경우 그에 대응하는 다기능 콘솔(Multi Function Console/MFC) 내의 전시 소프트웨어를 통하여 운용자의 명령을 처리한다. Fig. 4와 같이 데이터의 흐름을 보여준다.[12]

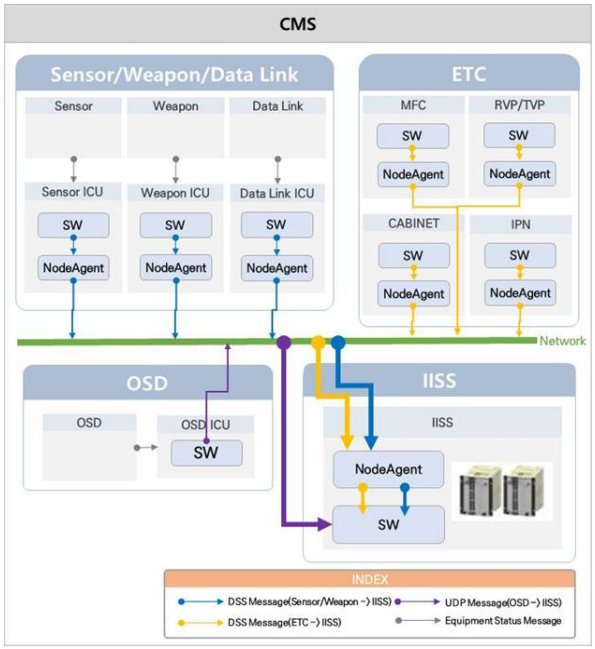

Fig. 4. Dataflow of CMS

각 연동 대상 장비와의 메시지 트랜잭션에서 처리되는 데이터들은 정보 저장(Integrated Interface Storage System/IISS) 소프트웨어에 수집되고 있으며 본 논문은 해당 장치의 토픽을 이용하여 운용자의 명령에 의해 정상적인 흐름의 데이터 집합들을 저장하고 머클 트리로 전환하여 목록화하였고 해당 데이터들을 추후 입력되는 시험 데이터들의 검증 데이터로 활용하였다.

3.2 Data Collection and Validation

검증을 용이하게 수행하기 위해 별도의 소프트웨어를 구현하고 해당 소프트웨어에서 IISS의 토픽을 수신하여 데이터의 수집을 수행하였다. 구현한 소프트웨어는 크게 수집 메서드와 검증 메서드 2가지 파트로 구현하였고 사용자에 의해 수집 기능이 수행될 때 IISS 토픽을 통해 데이터를 수신 시 메시지 원형 형태의 문자열로 저장하고 저장된 문자열 데이터를 라인 단위로 잘라서 노드를 구성 및 머클 트리로 변환하여 검증 데이터로써 저장하였다. 검증 메서드를 수행할 때도 마찬가지로 머클 트리화 절차를 수행하고 최종적으로 두 데이터의 머클 루트를 비교하고 불일치가 발생했을 시 노드 단위로 비교하여 불일치가 발생한 데이터를 확인하였고 여러 시나리오를 구성하여 해당 절차를 다회 수행 후 검증율을 측정하였다.

3.3 Expectation Effectiveness

본 논문에서 제시하는 전투체계 데이터 필드 단위의 무결성 검증을 통한 기대효과는 다음과 같다.

- 메시지 필드 불일치 검출에 따른 기능 오류 방지
- 소프트웨어 시험 단계에서의 절차 간소화
- 개발 유지보수 단계에서의 오류 식별 효율화

기 작성된 검증 데이터를 기반으로 개발 진행 시 메시지 필드 단위의 변경이나 잘못된 데이터 처리에 의한 기능 오류를 사전에 방지하고 추후 시험 단계에서도 수많은 필드 단위의 데이터를 확인하면서 시험을 수행할 필요 없이 입력행위에 대한 데이터와 검증 데이터의 일치 여부를 확인하여 소프트웨어의 메시지 처리 기능에 관한 시험을 간소화할 수 있다. 또한, 개발 유지보수 단계에서의 소프트웨어에 대해서도 사전에 정의한 정상 시나리오의 검증 데이터와 유지보수 필요 소프트웨어의 입력 로그의 비교를 통해 필드 단위의 오류 식별이 가능하다.

III. The Proposed Scheme

1. Collection and Veritication Method

본 논문에서 제시하는 검증 방안은 전투체계 메시지를 수집하여 검증 데이터로 저장하는 수집 메서드와 수신된 메시지를 검증 데이터와 비교하여 분석하여 결과를 도출하는 검증 메서드로 구성되어 있다.

- 수집 메서드 : CMS 데이터를 수신하여 필드 단위로 문자열 변환 후 라인별로 해시화를 진행하고 해시화된 모든 문자열 라인을 머클 트리로 변환하여 파일로 저장한다.

Table 1. Data Structure for DSS

Message ID	DSS Topic ID
Structure	DSS Data Name
Field Name#N	N Structure Fields
Field Value#N	N Structure Field Values

Table. 1은 DSS를 통해 전투체계 내에 유통되는 데이터의 형태이다. 수집 메서드는 위의 데이터를 해시화된 여러 라인의 문자열과 해당 문자열를 변환한 머클 트리의 머클 루트를 Table. 2와 같이 검증 데이터 파일로 관리한다.

Table 2. Stored Data Structure

Scenario ID	Collected Data Management
Merkle Root#N	Transformed Merkle Root#N Hash Values
Hashed Field#M	Transformed Field#M Hash Values

● 검증 메서드 : CMS 데이터를 수신하여 머클 트리로 변환한 후 검증 데이터의 머클 루트를 비교하여 검증을 수행한다.

2. Record CMS Message Transaction Scenario

본 논문에서 제시하는 검증 방안은 IISS 토픽을 이용해 전투체계 데이터를 수집하고 머클 트리로 관리하여 검증 데이터로 저장하는 구조로 설계하였다. 전투체계 전체 소프트웨어들은 자신들의 송수신 메시지를 IISS 토픽으로 송신하고 있기 때문에 해당 토픽을 수신 처리하여 구성할 경우 기존 소프트웨어의 수정 없이 제안하는 검증 방안을 기존 전투체계에 확장 구성이 가능하다.

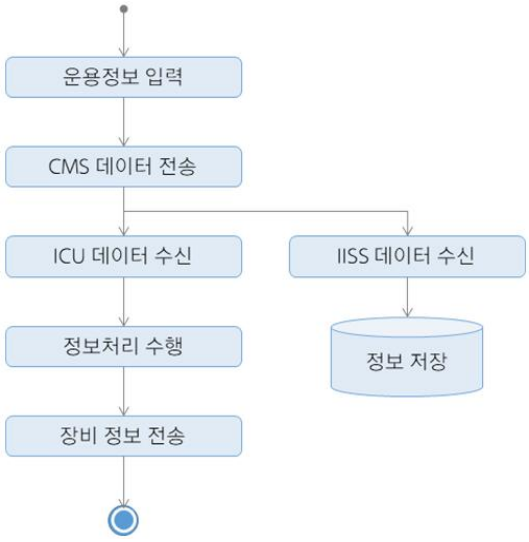

Fig. 5. Procedure of CMS

함정 전투체계에서 연동 대상 장비에게 명령을 전달하는 과정은 Fig. 5와 같다. 운용자는 전시 소프트웨어를 이용하여 연동 장비를 제어하는 데이터를 입력하고 전투체계는 해당 데이터를 체계 내에서 유통되는 구조의 데이터로 변환하여 ICU에게 전송한다. 이 과정에서 공용 토픽을 통해 IISS의 정보처리 소프트웨어에서 해당 체계 데이터를 저장하여 기록하고 추후 데이터 재생을 위해 사용되고 있다.

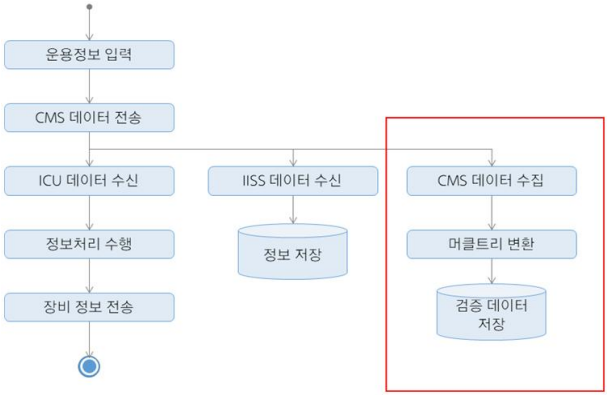

Fig. 6. Store Merkle Tree from CMS Data

Fig. 6과 같이 IISS 공용 토픽으로 데이터를 수신시 수집 메서드는 데이터를 문자열로 변환하여 해시화한 후 해시화된 문자열을 라인 단위로 해시화한 뒤 머클 트리로 변환한다. 이렇게 변환된 라인 해시값과 머클 트리를 검증 데이터로써 저장한다.

3. Merkle Tree for CMS Data

수집 메서드는 전투체계에서 데이터를 처리하는 일련의 과정을 시나리오화하여 관리하고 각 시나리오는 최소 1개 이상의 데이터 구조체 필드들이 해시화되어 저장된다.

Fig. 7. Convert CMS Data to Merkel Root

Fig. 7은 수집된 CMS 데이터를 데이터 필드 단위로 해시화하여 저장한 데이터의 모습이다. 각 라인별로 변환된 해시값은 머클 트리의 리프 노드로써 사용되며 각 해시값이 더해져 머클 루트를 완성한다. 또한 한 시나리오에서 N개 이상의 CMS 데이터가 처리되었을 때 각각의 구조체 머클 루트를 하나의 시나리오 안에 관리함으로써 한 시나이로 안에서 여러 토픽을 통한 메시지 처리 또한 가능하도록 설계하였다.

4. Verification Procedure

검증 메서드는 기본적으로 수집 메서드를 통해 시나리오 생성 후 수행된다. 운용자는 본인이 수행하고자 하는 연동장비 운용 절차를 수집 메서드를 통해 시나리오로 저장한다. 추후 전시 처리장치 및 연동 처리장치 내의 소프트웨어 변경 발생 시 메시지의 필드 불일치를 탐지하기 위해 저장된 시나리오를 선택 후 검증을 수행할 수 있도록 구성하였다. 검증 절차는 전투체계 데이터 수신시 1차적으로 해당 데이터를 머클 트리로 변환하여 머클 루트를 산출해낸다. 한 시나리오 내에 다수의 메시지를 처리해야할 경우 단일 메시지의 머클 트리를 저장한 후 수신될 메시지를 대기하고 메시지의 수신이 처리될 때마다 머클 트리를 생성하여 시나리오에 추가하고 수신이 종료될 때 메시지별로 머클 루트를 나열하여 저장한다.

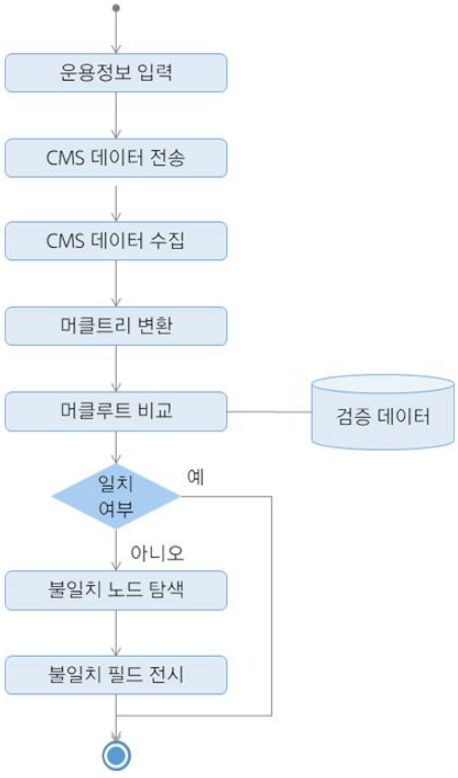

Fig. 8. Verificartion Procedure

Fig. 8과 같이 검증 절차는 수집이 완료된 테스트 데이터를 머클 루트로 변환하고 검증 데이터의 머클 루트와 비교를 수행한다. 비교 값이 일치할 경우 해당 테스트 데이터는 필드 단위의 오차가 없는 정상 시나리오를 수행했다고 판단되며 일치하지 않을 경우 필드 오차가 발생했다고 판단하여 테스트 데이터의 불일치 노드를 탐색한다. 불일치 노드의 탐색이 완료되면 최종적으로 불일치 필드를 전시하여 메시지 처리의 오류 사항을 Fig. 9와 같이 식별하였다.

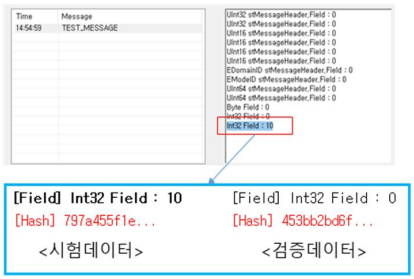

Fig. 9. Verification

IV. Software Evaluation

본 논문에서 제안한 메시지 데이터 필드 단위의 무결성 검증 방법이 전투체계에 적용 가능함을 확인하기 위해 다음과 같이 시험을 수행하였다. 검증 시험의 환경은 아래 Fig. 10 및 Table. 3과 같다.

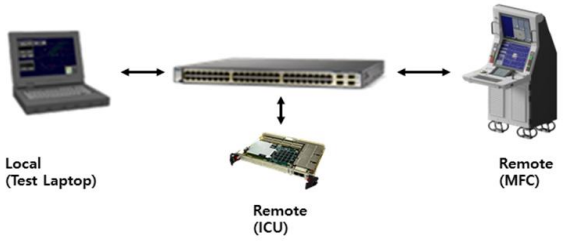

Fig. 10. Test Environment

Table 3. Test Environment

Equipment Name	CPU	Memory	OS
Test Laptop(Local)	Intel Core(TM) i7-4600M 3.6GHz	8GB	Windows10
ICU (Remote)	Intel Core i7 E610 @2.53GHz	4GB	RTST Linux
MFC (Remote)	Intel Xeon Gold 6142 2.6GHz	128GB	Windows10

테스트 환경은 실제 운용되는 사양과 동일한 ICU 및 MFC를 사용하였고 전투체계 데이터를 수집하기 위해 시험용 노트북을 추가 구성하여 검증을 수행하였다.

1. Verification Rate Measurement

임의의 전투체계 소프트웨어의 입력 데이터를 시나리오화하여 사전에 검증 데이터로 저장한 후 해당 소프트웨어를 종료 후 재실행하여 입력 상태를 초기화하였다.

Table 4. Test Scenario

Test Data	Verification Data
[10] [Int32] Speed : [1-100]	[10] [Int32] Speed : [20]
[10] [Type] Speed : [20]	[10] [Int32] Speed : [20]
[1-100] [Int32] Speed : [20]	[10] [Int32] Speed : [20]

Table. 4는 검증에 사용한 테스트 시나리오의 특정 데이터 필드의 구성이다. 10번 필드, Int32 타입의 Speed 필드에 20 데이터가 입력된 검증 데이터를 저장하고, 시험 데이터의 필드 데이터, 필드 타입, 필드 순서를 랜덤 함수를 통해 변경하여 전송하도록 하여 케이스마다 100회 수행하여 검증률을 측정하였다.

Table 5. Data Field Verification Rate

Test Case	Test Count	Verification Rate
Case 1. Changing Value of Field	100/100	100%
Case 2. Changing Field Type	100/100	100%
Case 3. Changing Field Order	100/100	100%

2. Validity Test

본 논문에서 제안하는 검증 방식은 별도의 분석 장치를 연결하여 시험 단계에서 검증 데이터를 기반으로 시험을 수행하여 시험 입력 데이터의 불일치를 탐지하는 것을 목표로 한다. 이를 위해 시험 수행 단계에 소요되는 시간 대비 검증 시간이 크게 소요되지 않음을 확인하기 위해 전투체계 메시지의 데이터 필드의 개수 별로 검증 소요 시간을 측정하였다.

Table 6. Processing Time by Field Count

Number of Fields	Time(s)
10000	0.130
30000	0.443
50000	0.897
80000	1.269
100000	1.334

전투체계에서 유통되는 정보는 유기적이면서도 많은 데이터를 처리하기 때문에 하나의 메시지 구조 안에 최대 80000여 개의 필드를 처리하는 메시지도 존재한다. 이런 경우는 하나의 메시지 안에 배열 형식의 필드를 운용하여 정보 처리를 수행하며 시험 단계에서 해당 필드들의 유효성은 일일이 값을 대조하면서 결과를 확인하기 어렵기 때문에 보통 정상적으로 처리되는 운용 동작이 전시되는 화면을 육안 검사하는 방식으로 시험이 이루어진다. Table. 6은 전투체계 최대 필드 80000여 개를 커버하는 수치를 시험하기 위해 최대 필드 개수를 100000개를 최대치로 설정하여 검증을 수행하였고 최대치일 때 약 1.3초가 소요되어 시험 수행 시 검증 결과를 확인하는 데 소요되는 시간이 크게 소요되지 않음을 확인하였다.

3. Resource Monitor

전투체계의 시험 환경은 다양한 상황에서 유기적으로 운용되어야 하기에 시험 환경이 제약적일 수 있다. 제안하는 검증 방식은 기본적으로 별도의 랩탑 PC를 이용한 방법이지만 시험 환경의 제약사항으로 랩탑 PC를 이용하지 못할 수 있기 때문에 MFC 내에 동시 운용할 경우를 가정하여 리소스의 사용량을 측정했다.

Table 7. Compare Resource to CMS Software

Resources	Exist Software	Proposed Software
Maximum of CPU(%)	3.8	8
Maximum of Memory(kb)	31748	69804
User Objects	2269	44
GDI Objects	456	63

Table. 7은 전투체계 임의의 전시 소프트웨어와 제시한 검증 소프트웨어의 리소스 소모량을 비교 분석한 결과이다. 비교 결과 CPU 및 메모리의 최대 사용량이 기존 전시 소프트웨어 대비 검증 소프트웨어가 약 2배 이상 높게 측정되었다. 하지만 해당 리소스는 각 소프트웨어의 최대 리소스 사용량을 측정한 결과이며 검증 소프트웨어의 경우 최대 필드 데이터를 머클 트리로 변환하는 과정에서 소모되는 리소스였으며 최대치의 메시지를 변환 처리하여도 약 1초 내로 처리가 완료되기에 기존 전시 소프트웨어와 동시 운용 시 무리없이 동작함을 확인하였다.

또한, 전투체계는 다수의 전시 소프트웨어가 동시 운용되기에 사용자 개체 및 GDI 개체의 사용량이 많을 경우 콘솔 자체의 그래픽 처리 능력에 부하가 발생한다. 제안하는 소프트웨어는 사용자 개체 부분에서는 타 전시 소프트웨어 대비 약 0.2%, GDI 개체 부분에서는 약 14% 수준의 사용량을 가지며 이는 다수의 전투체계 소프트웨어와 동시 운용하기에 무리가 없는 수치로써 함정 전투체계에 적용 가능함을 확인하였다.

V. Conclusions

전투 관리체계는 연동 대상 장비와의 상호작용을 메시지 필드를 사용하여 수행하므로 응용 소프트웨어 간의 메시지 필드의 무결성 검증은 매우 중요하다. 연동 대상 장비 중에는 함정의 운항에 필요한 정보나 교전을 수행하기 위한 정보를 관리하는 장비가 존재하므로 체계 내의 원활한 기능수행을 위해 메시지 필드는 운용 설계에 맞게 처리되어야 한다. 현재 소프트웨어의 기능 수행을 확인하기 위해 수많은 개발 프로세스와 시험 단계를 적용하여 소프트웨어의 정상 동작 여부를 확인하고 있지만, 소프트웨어의 개선, 예측하지 못한 휴먼 에러에 의한 데이터 처리의 오류가 발생할 수 있다.

본 논문에서는 메시지 필드 단위의 불일치로 인한 소프트웨어의 오류, 나아가서는 연동 장비의 오류를 바로잡을 수 있는 하나의 도구로써 머클 트리를 이용한 전투체계 데이터 필드의 무결성 검증 방안을 설계 및 구현하였고 검증 기능에 대한 수행 능력 확인, 검증에 따른 영향 분석을 통해 제안하는 검증 방안이 전투체계 내에 적용 가능함을 확인하였다.

해당 검증 방안은 기본적으로 시험 시나리오를 수동으로 저장하여 추후 소프트웨어 시험 수행 시 저장된 시험 시나리오를 통해 검증이 수행된다. 하지만 이런 방식은 각 개발자들이 검증용 시험 시나리오를 개별로 저장하여야 하며, 시험 수행 시에도 검증 시나리오를 선택하여 수행하도록 설계 되어 있다. 추후 연구 과제로는 이러한 불편 사항들을 개선하여 검증 시나리오를 DB화하여 통합 관리하도록 설계하고 메시지 처리시 자동으로 DB를 통해 검증을 수행하는 전투체계 트랜잭션 무결성 검증 자동화에 대한 연구를 진행할 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 Korea Association of Defense Industry Studies, 16(2), 237-258.

[2] Hwan-Jun Choi (2022). A Study on the Software Standardization and Simulator Design for Efficient Reliability Test in Combat System. Journal of the Korea Society of Computer and Information , 27(12), 151-159.

[3] Sang-Min Kwon, & Seung-Mo Jung (2018). Virtualization based high efficiency naval combat management system design and performance analysis. Journal of the Korea Society of Computer and Information , 23(11), 9-15.

[4] Young-Dong Heo (2020). A Study on the Standardization of System Support Software in the Combat Management System. Journal of the Korea Society of Computer and Information , 25(11), 147-155.

[5] Juwon Lee (2017). Development of Message Define & Management System based on Distributed Processing Environment for Naval Combat Systems. KIISE Transactions on Computing Practices, 23(12), 670-676, 10.5626/KTCP.2017.23.12.670

[6] Gautham Sekar, Soumyadeep Bhattacharya. (2016). Practical (Second) Preimage Attacks on the TCS_SHA-3 Family of Cryptographic Hash Functions. JIPS(Journal of Information Processing Systems), 12(2), 310-321.

[7] P. S. Almeida, C. Baquero, N. Preguiça, and D. Hutchison, “Scalable bloom filters,” Information Processing Letters, Vol. 101, No. 6, pp. 255-261, Mar. 2007.

[8] Becker, Georg. "Merkle signature schemes, merkle trees and their cryptanalysis." Ruhr-University Bochum, Tech. Rep 12 (2008): 19.

[9] Niaz, Muhammad Saqib, and Gunter Saake. "Merkle hash tree based techniques for data integrity of outsourced data." GvD 1366 (2015): 66-71.

[10] Byeong-Wan Lee (2024). Enhancing Installation Security for Naval Combat Management System through Encryption and Validation Research. Journal of the Korea Society of Computer and Information , 29(1), 121-130.

[11] Yong-Gyu Jung (2022). A Study on the Verification of Integrity of Message Structure in Naval Combat Management System. Journal of the Korea Society of Computer and Information , 27(12), 209-217.

[12] 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.

Authors

Young Choi received a Bachelor of Business Administration degree in IT Management from Woosong University, Korea, in 2014. He is currently working in Hanwha Systems Co. from 2021.

He is interested in Naval Combat System, Tactical Data Link System and so on.