Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.26, NO.4, Aug. 2016 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2016.26.4.929

안전한 해사클라우드 환경을 위한 SH-Tree 기반의 데이터 동기화 기법 제안*

이 동 혁,[1†] 박 남 제[1,2‡]
[1]제주대학교 일반대학원 컴퓨터교육전공, [2]제주대학교 초등컴퓨터교육전공

A Proposal of SH-Tree Based Data Synchronization Method for Secure Maritime Cloud*

Donghyeok Lee,[1†] Namje Park[1,2‡]
[1]Dept. of Computer Education, Graduate School, Jeju National University
[2]Dept. of Computer Education, Teachers College, Jeju National University

요 약

국제해사기구(IMO)의 e-Navigation 전략에 따라, 선박 및 육지간 안전하고 정확한 정보 전달을 지원하는 통신 인프라가 필요한 상태이다. 해사클라우드는 이러한 필요성에 따른 해양 통신 인프라의 개념으로 사용되고 있으며, e-Navigation 도메인 지원에 필요한 여러 요소들을 지원하고 있다. 안전한 해양 환경을 위해서는 해사 클라우드에 대한 연구와 함께 보안의 고려가 반드시 필요하나, 아직까지 해사 클라우드 보안에 대한 연구는 초기 단계이다. 본 논문에서는 해사 클라우드 서비스의 원활한 제공에 필요한 데이터의 안전한 동기화 기법을 제안하였다. 제안한 방식은 본 논문에서 제안한 SH-Tree 기반으로 선박간 동기화가 가능하며, 동기화 과정에서 정보 노출이 없다는 장점이 있다.

ABSTRACT

The IMO e-navigation strategy has requested a communication infrastructure providing authorized seamless information transfer between stakeholders. The Maritime Cloud is the term used to describe the concept of an infrastructure that support authorized, seamless information transfer, adding those elements, that are necessary to support the e-navigation domain. It is necessary to consider the study on maritime cloud security, but the study is still an early stage. In this paper, we propose a secure synchronization method for the maritime cloud services. The proposed method can be synchronize between the vessel based on the SH-Tree, and it has the advantage that there is no exposure information in the synchronization process.

Keywords: maritime cloud, cloud security, data synchronization

I. 서 론

국제 해사기구인 IMO에서는 선박운항과 정보기술을 융합하기 위하여 e-Navigation 전략을 수립한 바 있다. 과거에 선박사고의 대부분이 운항미숙과 과실 등 인적요인에 대한 해양사고가 많았던 바, 이러한 문제를 개선하기 위해 e-Navigation의 필요성이 크게 대두되고 있으며, 국내에서도 한국형 e-Navigation 구축을 목표로 진행중에 있다.

해사 클라우드는 e-Navigation에서의 주요 통신 기반 기술이다. 이는 일반적인 스토리지 클라우드와는 개념이 다르며, 해상 도메인의 다양한 시스템 간, 그리고 여러 통신 링크간 원활히 정보 교환을 가능하

Received(03. 18. 2016), Modified(1st: 06. 16. 2016, 2nd: 08. 02. 2016), Accepted(08. 03. 2016)
* 이 논문은 2016년 대한민국 교육부와 한국연구재단의 지원을 받아 수행된 연구임(NRF-2015S1A5A8018037).
† 주저자, bonfard@jejunu.ac.kr
‡ 교신저자, namjepark@jejunu.ac.kr(Corresponding author)

게 하는 목적으로 추진되고 있다.

해사 클라우드에 대한 연구는 아직 초기단계이며, 이에 대한 연구 가운데 정보보안에 대한 연구는 미진한 단계이다. 해사 클라우드는 기밀성, 무결성과 같은 기본적인 서비스는 제공하고 있으나, 위장공격, 정보교란, 권한 탈취, 서비스 거부 공격, 메시지 도청 공격, 메시지 위/변조 공격, 불법정보 접근, 내부자 공격 등과 같은 여러 공격에 취약할 수 있다[1].

e-Navigation은 편리한 도구가 될 수도 있지만, 보안에 취약하다면 큰 재앙이 될 수 있다. 해사 클라우드 시스템상의 보안적인 결함은 선박의 안전과도 직결되기 때문이다. 따라서, 해사 클라우드 보안에 대한 연구가 필요한 상황이다.

본 논문에서는 안전한 해사 클라우드 환경에 적합한 데이터 동기화 기법을 제안하였다. 제안한 방식의 특성으로, 새롭게 제안된 SH-Tree를 사용하며, 지오캐스팅을 기반으로 서버와 통신이 두절된 상태에서도 동기화가 가능하다. 또한, 델타 업데이트를 기반으로 최소한의 정보만 전달하므로 낮은 대역폭을 갖는다. 한편, 동기화 과정에서 데이터의 무결성을 보장할 수 있으며, 스토리지 절감을 위한 데이터의 중복 방지도 가능하다.

II. 관련 연구

본 장에서는 해사 클라우드의 개요 및 구성요소와 함께 클라우드 동기화의 연구 현황을 살펴본다.

2.1 해사 클라우드

2.1.1 개요

해사 클라우드는 이용가능한 통신 시스템을 통해 인가된 모든 해상 관련 당사자들 간의 효율적이고 안전하며, 안정적이고 원활한 정보 교환을 가능하게 하는 통신 체계이다.

해사 클라우드는 IMO의 전략에 따라 원활한 정보 전달을 지원하는 인프라의 개념을 설명하기 위해 제안되었다. IMO의 e-Navigation 전략에 따라 관련 당사자들 사이에 원활한 정보 전달을 제공하는 통신 인프라가 필요하였으며, 신뢰성 있고 상호 운용 가능한 서비스를 제공하기 위하여 해사클라우드의 필요성이 대두되었다. 특히 효율적이고 지속가능한 통신 인프라 체계의 확립이라는 점에서 의미가 크다.

해사 클라우드는 e-Navigation 아키텍처에서 통신수단으로서 활용되며, 여러 통신 링크 간에 원활하게 정보 전달이 가능하게 한다. 또한, 해상 관련 당사자들이 특정 통신 시스템 또는 채널을 선택하는 복잡성을 해결하기 위하여 통신 인프라로서의 게이트웨이를 통해 정보를 교환하게 한다.

2.1.2 해사 클라우드의 주요 요소

해사 클라우드의 주요 구성요소로서 해상 통신에서의 안전한 인증을 위한 해상 식별자 레지스트리(Maritime Identity Registry), 해사 서비스의 원활한 제공을 위한 해상 서비스 포트폴리오 레지스트리(Maritime Portfolio Registry), 사업자에 상관 없이 사용할 수 있는 메시지 허브인 해상 메시지 서비스(MMS)가 있다.

이러한 구성요소를 바탕으로 해사 클라우드는 인가된 해상 행위자 커뮤니티에 다른 해양 행위자가 접근시 식별 및 인증을 수행하고, 안전한 데이터 접속 환경을 제공해 주며, 원활한 로밍을 기반으로 위치탐지 관련 정보 서비스를 제공한다. Fig. 1.은 해사 클라우드의 주요 구성요소를 나타낸다.

해사 클라우드 상에서 선박들은 국제 상호 운용성 서비스인 해상 메세징 서비스를 추가로 사용하여 통신 링크간 원활한 로밍을 실현할 수 있다. 즉, 해사 클라우드를 통하여 상용화된 상업적인 데이터 링크에 의한 지오캐스트 MSI 서비스의 제공이 가능하며, 행위자는 자신의 위치 주변에 있는 지역에서 방송 또는 청취에 사용되는 통신 링크에 상관없이 논리적으로 해당 지역에서 방송을 하거나 청취할 수 있다.

본 논문에서는 효율적인 동기화를 위하여 지오캐스팅 채널을 활용한다. 즉, 여러 원인으로 해사 클라

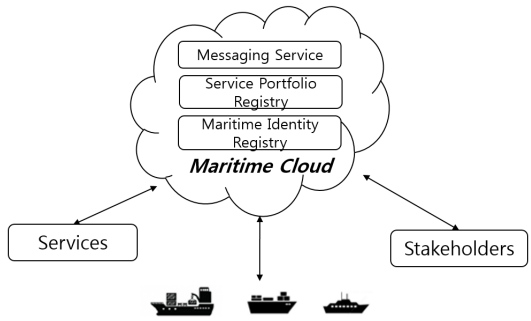

Fig. 1. The Core Elements of The Maritime Cloud[3]

우드 서버와 오프라인 상태에 있더라도 지오캐스팅을 통하여 주변 선박이 소유한 SH-Tree를 기반으로 최신 버전의 데이터로 갱신할 수 있다.

2.1.3 해사 클라우드의 보안 위협

해사 클라우드는 e-Navigation 통신 기반 기술의 핵심으로, 전 세계를 대상으로 통신이 연결된다고 볼 수 있다. 이러한 특성에 따라, 해사 클라우드에서의 보안은 필수적으로 고려되어야 한다.

해사 클라우드는 여러가지 다양한 통신 인프라를 활용하여 데이터가 전달된다. 즉, 단순히 인터넷 뿐만이 아니라, 여러 다양한 모든 통신 채널이 모두 포함될 수 있다. 이러한 특징은 해사 환경에서의 통신 가용성 측면에서 큰 장점을 준다.

그러나, 이러한 이면에는 보안상 큰 위험성이 존재한다. 데이터의 전달을 위하여 여러 검증되지 않은 채널을 통할 수 있고, 전달 과정에서 수많은 전달자가 존재할 수 있다. 이러한 점은 데이터 전송에 대하여 메시지 위/변조 공격의 가능성이 종래의 인터넷 환경보다 더욱 넓어질 수 있음을 시사한다.

이러한 점은 선박의 안전과 직결될 수 있는 위험이 존재한다. 만약, 악의를 가진 자가 임의로 MSI 등 여러 해사 관련 메시지에 대한 위/변조 공격을 수행할 경우, 선박은 잘못된 해사 데이터를 수신하여 안전한 운항에 큰 지장을 초래할 수 있다.

또한, 여러 다양한 채널의 활용에 따라 더욱 강한 무결성의 보장이 필요하다. 데이터의 무결성은 선박 운항에 반드시 필수적인 요소이며, 여러 채널로 통신하는 과정에서 데이터의 손실이 발생하면 즉시 판단이 가능해야 하고, 갱신할 수 있는 구조가 필요하다.

2.2 기존의 데이터 동기화 기법

2.2.1 개요

클라우드 환경에서 데이터 동기화는 널리 쓰이는 요소기술이며, 이는 정보 교환을 바탕으로 여러 장치간 가지고 있는 데이터를 동일하게 일치시키는 것을 목적으로 한다.

지금까지 동기화 방식에 대한 여러 연구가 진행되어 왔다. PDA와 PC간의 데이터 동기화를 시작으로, 디바이스 간의 동기화를 위하여 Palm OS에서 동작하도록 만든 HotSync, Windows CE기반에서 동작하는 Active Sync 등이 있다. 그러나, 이러한 방식들은 단일 플랫폼에 종속적이고, 동기화 시에 전체 데이터를 전송하는 특징이 있어 클라우드 환경에서는 효율적이지 않다. 이러한 단점을 보완하기 위해 파일 유사도 중복 체크, 델타 업데이트, 멀티 디바이스 동기화 지원 등을 지원하는 다양한 연구가 진행되었다[7,8,9,10,11].

2.2.2 H-Tree 기반 동기화 방식

H-Tree(해쉬 트리)는 1979년 Merkle에 의해 처음으로 제안되었다[19]. 이 방식은 데이터를 각 블록 단위로 분할한 뒤, 각각의 블록에 해쉬값을 수행하고, 이를 리프 노드에 입력한 후 단일 해쉬값이 만들어 질 때까지 트리 형태로 반복하여 해쉬를 하는 이진 해쉬트리 형태로 구성한 것이다. 이 경우 루트 해쉬 값은 서명으로 사용할 수 있다. 이러한 방법은 주로 데이터의 무결성 확인에 사용된다. Fig.2 는 H-Tree의 구조를 나타낸다.

H-Tree는 동기화에 응용하여 사용될 수 있다. 이는 강력한 무결성이 요구될 때 주로 사용되며, 모든 블록 데이터에 대한 해쉬값을 각각 비교함으로써, 만약 값이 다를 경우는 해당 데이터를 갱신하는 방식으로 데이터의 동기화를 완료하는 방식이다.

그러나, H-Tree는 무결성을 제외하고는 기본적으로 보안에 대한 특별한 고려는 하지 않고 있다. 예를 들어 해쉬값의 일부를 임의로 변경하였을때는 무결성 여부의 판단이 가능하지만, 원본 데이터의 일부가 변조되었을 경우, 그 상태 그대로 해쉬값을 재구성하여 전달한다면 별도의 원본 서명값을 확보하지 않는 이상 무결성 확인이 불가능하다는 단점이 있다. 아울러, 데이터 삽입/삭제 시 델타 업데이트 처리가

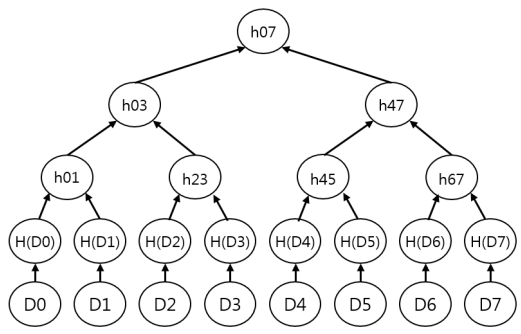

Fig. 2. H-Tree[19]

지원되지 않는다는 단점도 존재한다.

본 논문에서는 이러한 H-Tree의 단점을 보완한 SH-Tree를 제안한다. 제안한 방식은 트리 자체만으로 데이터의 변경 여부 확인이 가능하다는 장점이 있으며, 원본에 대한 위/변조, 무결성이 동시에 요구되는 해사클라우드 환경에 적합하다. 또한, 델타 업데이트가 가능하여 대역폭을 최소화하면서 최소한의 시간 내 동기화 처리가 가능하다.

III. 새로운 동기화 기법 제안

본 장에서는 데이터 동기화 방식의 설계를 위한 기술적 요구사항을 살펴보고, 데이터의 안전한 동기화 기법을 제안한다.

3.1 제안 기법 개요

제안한 방법은 해사 클라우드 환경에서 SH-Tree를 기반으로 데이터를 안전하게 동기화하는 방법에 관한 것으로, 해사 클라우드 서버 내의 데이터와 각각의 선박이 가지고 있는 데이터가 완전히 일치하게 동기화 하는 것을 목적으로 한다.

동기화는 특정 지리적 범위 내 선박을 대상으로 하며, 범위 내에 있더라도 인가되지 않은 선박은 동기화 대상에서 제외된다. 또한, 대역폭 최소화를 위한 델타 업데이트를 지원해야 하며, 데이터 전송 과정에서의 위/변조 방지가 가능해야 한다.

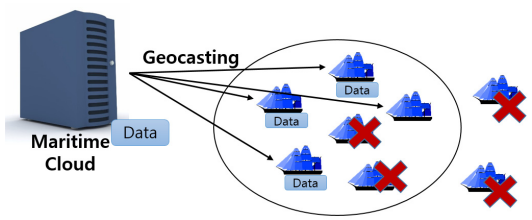

Fig. 3. Overview of Proposed Method

3.2 세부사항

본 절에서는 해사 데이터 동기화를 위해 제안한 기법의 구체적인 수행 절차를 설명한다.

3.2.1 표기법

본 논문에서 제안하는 기법의 설명을 위해, Table 1.에서 약어를 설명한다.

Table 1. Notation

Abbreviation	Explication
Dn	n-th Data Block
Mn	Metadata of Dn
Vn	Last Version of Dn
K_p	Pre-Shared Key
H(·)	Hash Function
E(·)	Encryption

3.2.2 SH-Tree

본 논문에서는 해사 클라우드의 안전한 동기화를 위해 H-Tree를 변형한 SH-Tree(Secure Hash Tree를 제안한다. 이는 평문을 암호화된 데이터와 메타정보로 분할하여 해쉬트리를 구성하는 것이 특징이며, 데이터 블록의 가변 사이즈 관리가 가능하다. 또한, 동기화된 정보와 메타정보 내의 값을 비교함으로써 데이터의 무결성, 위/변조여부의 확인이 가능하다. Fig.4.는 SH-Tree를 나타내며 Table.2.는 메타정보 M이 담고 있는 항목들을 나타낸다.

메타정보는 데이터 블록의 사이즈인 S_n, 버전 정보를 나타내는 V_n, 타임스탬프 ts, 평문 데이터 블록의 해쉬값을 나타내는 h(Dn)으로 각각 구성된다. 여기에서 S_n값은 가변 사이즈의 관리를 위해 필요하다. 만약, 데이터 블록에 값이 추가되어 삽입되었을 경우는 아래와 같이 S_n의 갱신이 필요하다. 기존 해

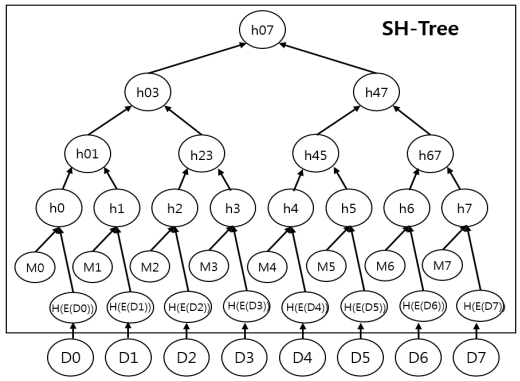

Fig. 4. SH-Tree

Table 2. Contents of Metadata

Abbreviation	Content
S_n	Size of n-th data
V_n	Version of n-th data
ts	Timestamp
h(Dn)	Hash value of n-th data

시트리 방식에서는 데이터 삽입 시 해당 블럭 이후의 모든 블록에 대한 해쉬값이 변경되는 문제점이 존재하였다. 그러나, SH-Tree에서는 메타정보상에 블록 사이즈를 관리하는 방식으로 가변 사이즈를 처리할 수 있다. 즉, 데이터의 삽입이 일어나더라도 해당 데이터 블록의 해쉬값 및 메타정보 내의 사이즈 값만 변경되고, 다른 데이터 블록은 영향을 받지 않는다.

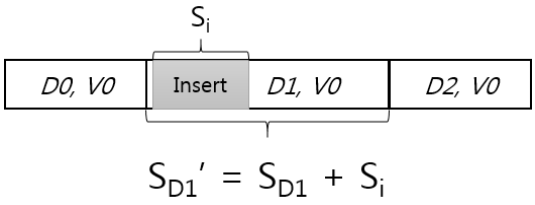

Fig. 5. Size of Block Data

3.2.3 SH-Tree의 구성 및 전달단계

데이터 암호화 단계는 데이터를 임의의 사이즈의 블록으로 분할한 뒤, 해당 블록에 대한 암호화를 수행하고, 메타정보와 함께 SH-Tree를 구성하는 단계이다. 원본 데이터를 블록 단위로 처리하는 이유는 효율성 및 부분 변경사항 검출에 목적이 있다. 데이터의 일부가 변경된 경우는 변경된 부분의 데이터 블록만 동기화를 수행하면 되며, 전체 데이터를 다시 동기화 할 필요가 없기 때문이다.

n 번째 블록에 대한 암호화 키의 구성방법은 다음과 같다. 아래의 식에 따라, 암호화 키는 각각의 데이터 블록 단위로 달라지게 된다.

$$K = K_P \oplus H(IDX_{Dn} + V_{Dn})$$

위와 같이 초기 키값인 K_P를 기반으로, 데이터 블록의 인덱스와 버전정보를 합하여 XOR처리하는 것으로 데이터 블록의 암호 키를 생성할 수 있다. n번째 데이터 블록의 인덱스값인 IDX_{Dn}은 아래의 식으로 구할 수 있다.

$$IDX_{Dn} = \sum_{i=1}^{n} S_i$$

Fig. 6.은 데이터 블록의 암호화 방법을 나타낸다. 각각의 블록은 버전정보가 별도로 관리되고 있으며, 이는 블록의 변경사항이 순서대로 이루어지는지를 판별하고, 필요시 데이터를 이전버전으로 복원하기 위한 목적으로 관리된다.

데이터의 암호화가 끝나면, 데이터 블록으로부터 메타정보를 추출하고, SH-Tree를 구성한다. SH-Tree는 지오캐스팅 등 다양한 채널을 통하여 선박에 전달 가능하며, 선박은 SH-Tree의 값을 기반으로 동기화를 수행할 수 있다.

SH-Tree의 전달은 인가된 각 선박들을 대상으로 Fig.7.과 같이 지오캐스팅을 기반으로 SH-Tree를 배포한다. 이는 불특정 다수의 선박을 기반으로 암호화된 상태의 SH-Tree를 방송하며, 인가된 선박만

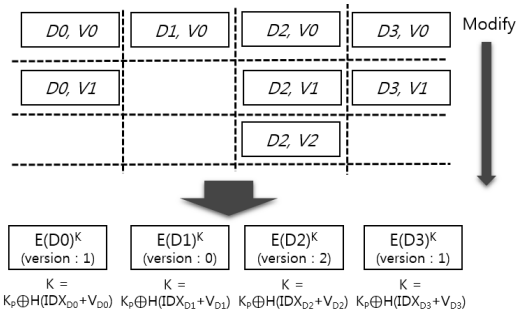

Fig. 6. Encryption of Block Data

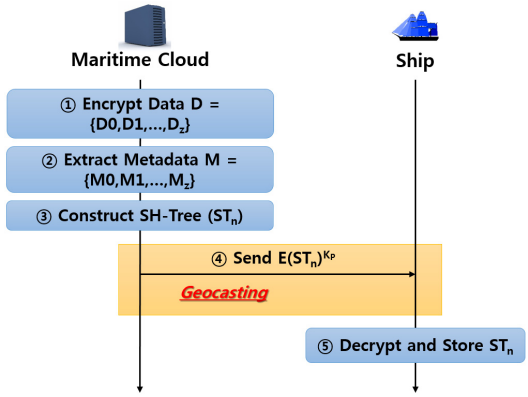

Fig. 7. SH-Tree Delivery Phase

이 SH-Tree를 복호화하여 수신할 수 있다.

SH-Tree의 구체적인 생성 및 전달 절차는 아래와 같다.

① 서버는 각 데이터 블록에 대응하는 키 K를 생성하고, 이를 기반으로 암호화를 실시한다.
② 서버는 각 데이터 블록에 대응하는 메타데이터를 생성한다.
③ 서버는 암호화된 데이터 블록과 메타데이터를 기반으로 SH-Tree를 구성하고, 이를 ST_n로 한다.
④ 서버는 지오캐스팅을 통하여 불특정 다수의 선박에 K_P로 암호화된 ST_n를 배포한다.
⑤ 선박은 데이터 수신 후 복호화하여 ST_n을 저장한다.

3.2.4 데이터 동기화 단계

데이터 동기화 절차는 데이터 SH-Tree를 기반으로 데이터 변경사항을 검출하고 실제로 변경된 부분에 대한 데이터를 수신하는 단계이다. Fig.8.은 이전에 수신한 SH-Tree와 새로 수신한 SH-Tree' 간 차이점을 발견한 모습이다. 그림에서는 데이터 블록 D1과 D3이 변경된 것이 감지되었을 경우를 나타내며, 이러한 경우 해당 블럭에 해당하는 상위 트리에 대한 모든 값이 변경된다. 데이터 블록의 해쉬값을 트리형태로 관리하는 것은 선형적인 목록값으로 가지고 있는것 보다 변경사항 검출 속도를 증가시킨다는 장점이 있다. 데이터 블록의 해쉬를 리스트 형태로 가질 경우 동기화를 위해 전체 해쉬 블록에 대한 선형 검색이 필요하므로 변경사항 검출시 O(n)의 시간이 소요되나, 트리로 구성하였을 경우는 $O(\log_n)$의 시간으로 변경여부의 검색이 가능하기 때문이다.

클라이언트는 변경된 블록을 서버에 요청하며, 서버는 해당 요청된 데이터 블록에 대하여 데이터베이스 및 스토리지 서버로부터 동일한 해쉬값을 가지고 있는 파일이 있는지를 찾는다. 이 경우, 매핑 테이블이 사용된다. 실제 파일은 클라우드 스토리지에 저장되어 있으며, 매핑 테이블은 해당 스토리지와 실제 파일의 해쉬값을 연결해 준다. 또한, 매핑 테이블은 데이터의 중복 제거 기능을 제공해 주어 데이터 사이즈 측면에서 크게 효율적인 스토리지 관리가 가능하다. Fig.9.는 스토리지와 연결된 데이터베이스의 매핑 테이블 구조를 나타낸다. 해당 그림에서는 동일한 값을 가진 데이터 블록에 대하여 스토리지에서 복수로 보관하지 않고, 단 하나의 데이터 블록만 보관하여 중복 제거가 적용될 수 있음을 보인다.

Fig.10.은 데이터 동기화 절차를 나타낸다.

① 클라이언트는 이전에 수신한 ST_{n-1}과 새로 수신한 ST_n을 비교하여 어느 블록에 대한 변경사항이 있는지를 감지한다.
② 클라이언트는 변경사항이 있는 블록의 집합을 서버에 요청한다. 이때, 요청값은 각 데이터 블럭에

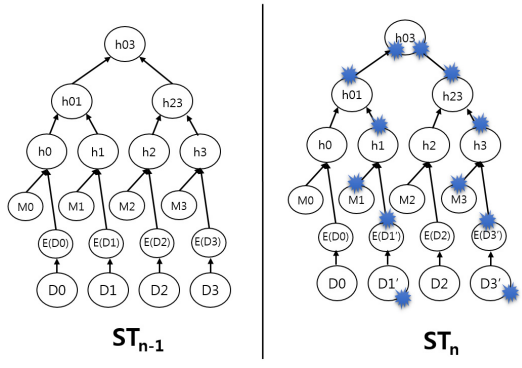

Fig. 8. Detection of Changes

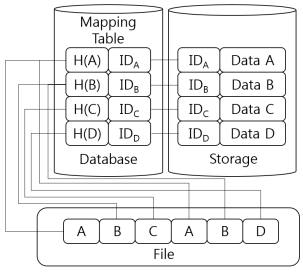

Fig. 9. Mapping Table for De-duplication

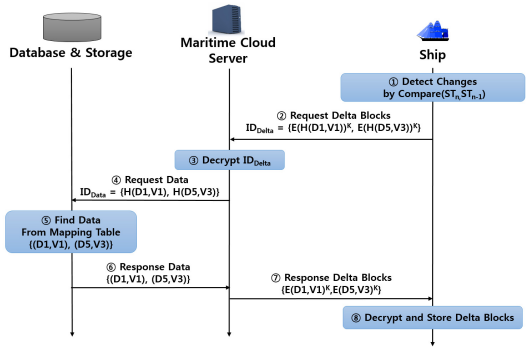

Fig. 10. Data Synchronization Phase

해당하는 해쉬의 암호화된 값으로 한다.

③ 서버는 요청된 블록에 대하여 복호화 후 데이터 블럭의 해쉬값을 얻는다.

④ 서버는 데이터베이스 및 스토리지 서버에 해당 블럭의 해쉬값에 대응하는 데이터를 요청한다.

⑤ 데이터베이스의 매핑 테이블을 기반으로 스토리지 내의 실제 데이터를 얻는다.

⑥ 데이터베이스 및 스토리지 서버는 클라우드 서버 측으로 실제 데이터를 전달한다.

⑦ 클라우드 서버는 해당 데이터를 암호화하여 선박 측에 전달한다.

⑧ 선박은 수신한 데이터를 복호화하고, 저장하여 동기화를 완료한다.

3.2.5 통신 두절시의 데이터 갱신

해사 클라우드 환경의 특성상, 선박측과 통신 두절시의 환경도 같이 고려되어야 한다. 만약 통신 두절 사유로 인해 서비스가 중지된다면, 선박의 안전이 크게 위협받을 수 있기 때문이다. 기본적으로 통신 두절시는 서버상의 최신 정보를 전체 선박에 배포하는 것은 불가능하다. 이러한 경우, 서버의 가장 최신 정보가 아니더라도, 무선 통신이 가능한 인근 선박간 소유한 데이터 가운데 가장 최신 정보로 갱신되는 방식으로 동기화가 가능해야 한다.

따라서 본 논문에서 제안된 방식은 해사 클라우드의 서버 측 뿐만 아니라, 인가된 그룹 내의 다른 선박에 대해서도 요청이 가능하다. 즉, 서버와 통신이 두절된 경우라 할지라도 선박간의 무선 통신이 가능한 경우는 동기화 진행이 가능하다. Fig. 11.은 선박간에 동기화가 이루어지는 경우를 나타내고 있다.

기본적으로, 이러한 문제는 Ship A와 Ship B간에 어느 정보가 가장 최신 것인지에 대한 고려가 먼저 필요하다. 즉, 각 선박에서 가지고 있는 데이터 블록 가운데 일부는 Ship A가 최신 데이터일 수도 있으며, 일부는 Ship B가 최신 데이터일 수도 있다. 이러한 관점에서 예를 들어, Ship A가 일방적으로 Ship B의 데이터를 신뢰하고 동기화해서는 안되며, 반대의 경우도 마찬가지이다. 만약 지오캐스팅으로부터 전달되었던 클라우드 서버의 SH-Tree라면 최신것으로 판단하고 동기화를 진행하면 문제가 없으나, 선박간의 동기화 시는 두 선박 간 어느 선박의 데이터가 최신 정보인지를 서로 비교하여 판단하는 것이 중요한 관점이다.

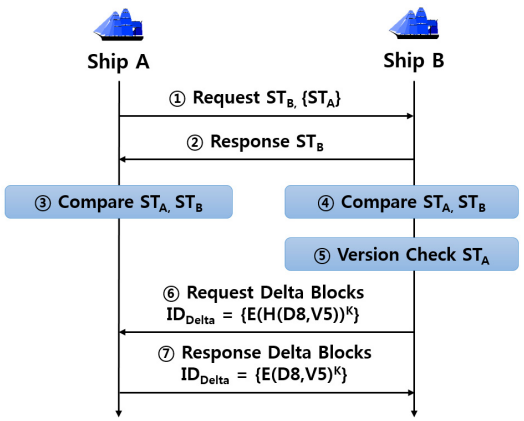

Fig. 11. Synchronize Data Between Vessels

본 논문에서는 데이터 블록에 대한 버전정보를 별도로 관리하여, 두 데이터 블록 간 어느 것이 최신정보인지에 대한 판단이 용이하도록 설계하였다.

Fig.11은 선박간 SH-Tree의 교환 결과 Ship A의 데이터 가운데 8번째 블록인 D8이 가장 최신인 것으로 Ship B가 판단하고 최신 값으로 갱신하는 과정을 나타내고 있다.

① Ship A는 Ship B의 SH-Tree를 요청함과 동시에, 자신이 소유하고 있는 SH-Tree인 ST_A를 전송한다.

② Ship B는 Ship A에게 자신이 소유한 SH-Tree인 ST_B를 응답한다.

③ Ship A는 ST_A와 ST_B를 비교하여 변경사항을 검출한다.

④ 동일하게, Ship B는 ST_B와 ST_A를 비교하여 변경사항을 검출한다. 여기에서는 8번째 블록인 D8인 검출되었다.

⑤ Ship B는 검출된 블록에 대한 버전값 비교를 통해, D8이 Ship A보다 낮은 버전임이 감지되었다.

⑥ Ship B는 Ship A로부터 (D8,V5)를 요청한다.

⑦ Ship A는 암호화된 (D8,V5)를 응답한다.

IV. 제안 기법 분석

본 장에서는 제안한 방식을 안전성 및 효율성 관점에서 분석하고, 기존의 해쉬 기반 동기화 방식과 비교한다.

4.1 안전성 분석

4.1.1 데이터 변조 공격

H-Tree 기반의 동기화 방식은 데이터 변조 공격으로부터 취약한 측면이 있다. 변조하고자 하는 데이터 블럭을 임의로 수정하고, 해당 블럭에 대한 해쉬값을 취한 후 모든 상위 해쉬 트리에 대하여 재해쉬 처리하면, H-Tree의 수신자는 해당 데이터가 변조되었는지를 확인할 수 없다. 만약 악의를 가진 자가 임의로 데이터를 수정하여 선박 측에 조작된 데이터를 전송한다면, 이는 선박의 안전에 큰 영향을 받게 될 수도 있다.

제안한 SH-Tree 방식은 변조되지 않았다는 부분에 대한 검증이 가능하다. SH-Tree의 메타정보상에 평문 데이터 블록의 해쉬값을 가지고 있으며, 동기화 이후 최종 상태에서 수신 데이터를 해쉬처리하고 이 평문 데이터 블록의 해쉬값이 일치하는지를 비교하면 실제로 변조되지 않았음을 알 수 있다.

즉, 공격자는 데이터를 임의로 변조하더라도 해당 평문 데이터 블록의 해쉬값을 생성할 수 없다. 이는 데이터 블록단위로 각각 다른 암호화 키를 가지고 있으므로 공격자는 해당 암호화된 데이터 블록으로 부터 평문을 복호화 할 수 없기 때문이다.

만약, SH-Tree상의 데이터를 무작위로 변경하였을 경우에도 실제 데이터의 블록 해쉬값과 SH-Tree상의 해쉬값이 일치하지 않으므로 정상적이지 않은 데이터임을 알 수 있을 것이다. 이러한 경우, 선박은 변조된 상황임을 감지하고 클라우드 서버로부터 다시 정확한 정보를 재요청하여야 한다.

4.1.2 무결성 보장

제안하는 방식은 H-Tree 방식과 동일하게 무결성을 보장해 준다. 데이터 통신 과정에서 손실 등으로 문제가 발생할 경우, 데이터의 해쉬값은 달라질 것이다. 이 경우, 최상위 트리값인 Root값 또한 달라지게 되며, 이를 기반으로 실제 데이터가 훼손되었는지를 안전하고 효율적으로 판단할 수 있다. 이러한 무결성 보장은 H-Tree에서 기본적으로 제공하는 속성이며, 제안한 방식은 이 장점을 동일하게 갖는다.

특히, 데이터의 손실이 어느 블록에서 발생하였는지를 구체적으로 알 수 있으며, 선박 측에서는 손실이 감지된 데이터 블록을 서버에 재요청하여 갱신해야 한다.

4.1.3 메시지 도청 공격

H-Tree 기반의 동기화 방식은 보안에 대한 별도의 고려를 하지 않고 있어, 메시지의 도청으로부터 안전하지 않다. 그러나, 제안한 방식은 전송되는 모든 데이터가 암호화되어 관리된다. 특히, 데이터 블록 단위로 각각 다른 키가 적용되어 암호화되므로, 공격자는 메시지 도청 공격을 실시할 수 없다. 또한, 이러한 특징은 데이터 블록이 어떤 반복적인 패턴을 가지고 있는지에 대한 추정도 어렵게 한다.

4.1.4 안전성 비교

해쉬 목록만으로 변경여부를 감지하는 Hash List 기반의 동기화 방식과 H-Tree 기반의 동기화 방식은 메시지 변조 공격에 대해서 안전하지 않다. 변조된 데이터 및 상위 트리의 값을 재해쉬 처리하면 실제로 데이터를 수신한 측은 데이터의 변조가 일어났는지 여부를 판단할 수 없기 때문이다. 그러나, 제안하는 방식은 SH-Tree 자체만으로 데이터의 검증이 가능하여 데이터 변조 공격을 방지할 수 있다.

한편, 무결성은 해쉬 기반 동기화 방식이 공통적으로 갖는 장점이며, H-Tree 및 제안한 방식 모두 공통적으로 가지고 있다. 한편, Hash List는 Root Hash 값이 존재하지 않으므로, 단일 Hash 값으로는 데이터의 무결성을 확인할 수 없고, 전체 해쉬값에 대하여 선형검색을 수행함으로써 데이터의 무결성 확인이 가능하다는 단점이 있다. 또한, Hash List와 H-Tree 방식은 보안에 대한 고려는 명시하지 않고 있어, 스니핑 공격으로부터 안전하지 않다. 제안한 방법은 암호화된 데이터 블록을 기반으로 SH-Tree를 생성하며, 데이터 전송 과정에서 암호화가 발생하고, 각각의 블록 단위로 다른 키를 갖게 되므로 스니핑 공격에서 안전하다는 장점이 있다.

Table 3. Security Comparision

	Hash List	H-Tree	SH-Tree (Our Method)
Modification	×	×	○
Integrity	△	○	○
Sniffing Attack	×	×	○

4.2 효율성 분석

4.2.1 델타 업데이트

제안한 기법은 델타 업데이트 기능을 지원하며, 이러한 점은 대역폭을 절약하며 효율적인 동기화 정보 전송이 가능하게 한다. 즉, 변경된 특정 블록에 대한 정보만 업데이트가 가능하다. 델타 업데이트를 극대화하기 위한 방안으로, SH-Tree는 가변 사이즈를 지원한다. 각 데이터 블록에 대응되는 메타정보에서 사이즈 값을 별도로 가지고 있으며, 이를 기반으로 구간별 사이즈를 정확히 산출이 가능하다. Fig.12.는 기존의 H-Tree 방식과 SH-Tree 방식을 비교하고 있다. H-Tree 방식은 동일 사이즈 내 데이터 변경에 대한 델타 업데이트는 지원 가능하나, 데이터 삽입, 혹은 삭제에 대한 부분은 지원하지 않는다. 이는 특정 블록에 데이터가 삽입되거나, 삭제되면 해당 블럭 이후 모든 블록에 대하여 해쉬값이 변경되기 때문이다. 즉, 특정 블럭에 대한 삽입이 일어나면 해당 블럭 이후의 모든 데이터 블럭에 대하여 해쉬값을 갱신해야 한다는 큰 단점이 있다.

그러나, SH-Tree의 경우는 사이즈 값이 명확히 관리되므로, 특정 블록에 대한 데이터의 삽입, 삭제가 일어나더라도 해당 블록에 대한 데이터만 변경하면 되므로 크게 효율적이다.

제안한 방식은 델타 업데이트의 지원에 따라 동기화를 위해 모든 파일을 업데이트하지 않아도 된다는 장점과 동시에, 동기화에 소요되는 속도 향상의 측면도 있다. 이는 원본의 부분 수정 후 동기화 시 전체 데이터가 갱신되지 않고, 변경된 블록에 대해서만 데이터가 갱신되므로 동기화 대역폭 절감 뿐 아니라 동기화 완료에 필요한 시간도 대폭 절감되기 때문이다.

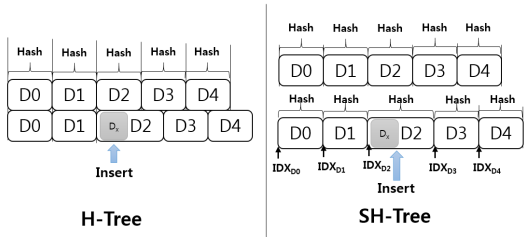

Fig. 12. Comparision of Each Method

4.2.2 통신 두절 관리

해사 환경의 특성상 천재지변 등 여러 사유로 서버와의 통신이 두절될 상황을 충분히 고려해야 하며, 이러한 경우는 인근 무선 통신이 가능한 선박간의 자원만으로 최신의 정보를 업데이트해야 한다.

제안한 방식은 서버와 데이터 통신이 두절된 상황이라 할지라도, 지오캐스팅 영역 내에 있는 주변 선박으로부터 최신 데이터의 동기화가 가능하다. 이는 SH-Tree의 메타정보 가운데 데이터 블록의 버전정보를 별도로 관리하기 때문이다. 각 블록의 버전 값 비교를 통하여 어느 블록이 최신 버전인지, 어느 블록을 기준으로 갱신 여부를 결정할 수 있다.

4.2.3 데이터 중복 제거

Hash List 방식과 H-Tree 기반의 동기화 방식은 데이터 중복 제거를 별도로 명시하지 않고 있다. 그러나, 이론상으로는 이러한 방식도 해쉬값에 대해서 매핑 테이블 기반의 중복제거 방식을 적용하면 데이터 중복 제거에 적용이 가능할 수 있다. 그러나, 이는 근본적으로 가변 사이즈의 데이터가 관리되지 않기 때문에 데이터 삽입/삭제 시 해당 블록 이후의 전체 블록에 대해서 데이터 변경이 일어나므로 실질적으로 데이터 중복 제거에 대한 효과를 크게 기대할 수 없다. 데이터 중복 제거는 동일한 내용을 가진 데이터 블록의 갯수가 최대한 많을수록 효율적이다. 이러한 관점에서는 데이터 삽입/삭제에 따라 데이터 변동폭이 큰 Hash List 및 H-Tree방식은 적합하지 않다고 볼 수 있다.

그러나, 제안한 방식은 가변 사이즈 방식의 데이터 블록 관리를 지원하며, 이러한 장점은 데이터 중복 제거에 있어서 큰 효과를 기대할 수 있다. 특정 블록에 대한 삽입과 삭제가 일어나도 해당 블록 이외의 블록에 대해서는 영향을 미치지 않고, 해당 블록에 대해서만 갱신하면 된다는 장점이 있어 데이터 중복 제거 효율을 크게 높인다.

4.2.4 변경사항 추출 성능

SH-Tree 기반의 추출 방법은 블록 아이디를 단순 목록으로 가지고 있는 Hash List 방식에 비해 검색성능이 뛰어나다. 각 데이터 블록에 대한 해쉬값을 Hash List 방식으로 관리할 경우, 변경사항 검

출시에 전체 데이터에 대한 선형 검색이 필요하므로 O(n)의 시간이 소요된다. 그러나, SH-Tree의 경우는 변경사항 검색에 $O(\log_n)$의 시간이 소요되며, 이는 종래의 H-Tree의 방식과 동일하게 장점을 가진다.

4.2.5 효율성 비교

Hash List 방식 및 H-Tree 방식은 델타 업데이트에 대하여 동일한 사이즈 내의 변경사항에 한해서만 일부 지원하고, 데이터의 삽입, 삭제 등에 대해서는 델타 업데이트를 지원하지 않는다. 한편, 제안한 SH-Tree 방식은 메타정보에 사이즈를 별도로 관리하여 델타 업데이트를 지원한다는 장점이 있다.

또한, 통신 두절시 Hash List 방식과 H-Tree 방식으로는 데이터 동기화를 추가적으로 진행할 수 없다. 어느 데이터 블록이 더 최신 버전인지 판단할 수 있는 근거가 없기 때문이다. 그러나 제안하는 SH-Tree 방식은 블록에 대한 버전 정보를 별도로 관리하므로, 가장 최신 버전에 대한 데이터를 동기화 하는 것으로, 통신 두절 시에도 인접 선박에 대한 데이터와 무선 통신으로 동기화가 가능하다.

한편, Hash List 및 H-Tree 방식을 데이터 중복제거 방식에 적용한다면, 특정 블록 이후로 모든 데이터가 변경되므로, 결국 해당 데이터 이후의 모든 데이터를 갱신해야 하므로 동기화 처리 비용은 선형 증가하게 된다. 그러나, 제안하는 방식은 특정 단일한 블록에 단 1회의 갱신 처리만 필요하다.

또한, 변경사항 감지에 있어서 Hash List 방식은 순차적으로 선형검색이 진행되므로, O(n)의 수행시간 내에 변경사항 검색이 가능하다. 한편, H-Tree 및 SH-Tree 방식은 동일하게 $O(\log_n)$의 시간에 변경사항 검출이 가능하다.

Table. 4.는 기 제안된 해쉬 기반의 동기화 방식과 본 논문에서 제안한 동기화 방식을 비교하였다. 제안한 방식은 대역폭 최소화를 위한 델타 업데이트를 지원하며, 통신 두절 상태에서도 서버와 동일한 것을 보장하지는 않더라도, 인접 선박의 정보를 기반으로 가능한 최신 정보로 갱신하는 방식의 동기화 처리가 가능하다. 또한, 클라우드 서비스에 주요한 요소인 중복 제거를 지원하고, 효율적인 변경 감지가 가능하여 해사 클라우드 환경에서의 동기화 서비스 제공에 적합하다고 볼 수 있다.

Table 4. Efficiency Comparision

	Hash List	H-Tree	SH-Tree (Our Method)
Delta Update	△	△	○
Loss of Connection	×	×	△
De-duplication	O(n)	O(n)	1
Chance Detection	O(n)	$O(\log_n)$	$O(\log_n)$

V. 결 론

해사 클라우드는 최근 새롭게 제안된 개념으로써 이에 대한 연구는 시작 단계에 있다. 이러한 해사 클라우드의 개발에 앞서 보안성에 대한 부분이 반드시 고려되어야 한다. 이는 통신상의 가용성 문제 뿐만 아니라 선박의 안전에 대한 부분과도 직결될 수 있기 때문이다. 따라서 본 논문에서는 안전한 해사 클라우드 환경을 위한 데이터 동기화 방법을 제안하였다. 제안한 기법의 특징으로 지오캐스팅 기반으로 동기화가 가능하며, 동기화 과정에서 데이터의 기밀성과 무결성이 보장된다. 또한, 메시지 위/변조 공격도 방지할 수 있으며, 델타 업데이트 및 데이터 중복 제거도 지원하여 매우 효율적이다.

이를 위해 먼저 2장에서 해사 클라우드의 개념 및 주요 요소를 살펴보고, 클라우드 컴퓨팅에서의 동기화 기술 연구 동향에 대하여 간략히 살펴보았다. 그리고 3장에서는 동기화 기법 제안에 앞서 기술적 요구사항을 먼저 살펴보고, 세부적인 동기화 절차를 설명하였다. 또한 4장에서는 안전성과 효율성 측면에서 분석을 진행하였다.

해사 클라우드에 대한 연구는 아직 초기단계이며, 해사 클라우드 보안에 대한 연구는 현재까지는 활발히 진행되지 않고 있다. 해사 환경에서의 보안은 선박의 안전과 직결된 만큼, 보안적인 측면에서의 연구도 동시에 진행되어져야 할 것이다.

References

[1] Gae Il An, Kwangil Lee, Byung Ho Chung, "Analysis of Cyber-Security Threat on Maritime Cloud proposed as Maritime Communication Framework," In Conference Proceedings of Korea

Information Science Society, pp.892-893, Dec. 2015.
〔2〕 Namje Park, Jungsoo Park, and Hyoungjun Kim, "Inter-Authentication and Session Key Sharing Procedure for Secure M2M/IoT Environment," International Information Institute(Tokyo) Information, 18(1), pp. 261-266, Jan. 2015.
〔3〕 https://imo.amsa.gov.au/iala-aism/e-nav/enav16/9-24.pdf
〔4〕 Namje Park and Marie Kim, "Implementation of load management application system using smart grid privacy policy in energy management service environment," Cluster Computing 17(3), pp. 653-664, Sep. 2014.
〔5〕 https://www.iho.int/mtg_docs/com_wg/SNPWG/SNPWG17/SNPWG17-9.3_An%20overview%20of%20the%20Maritime%20Cloud%20-%20input%20to%20IMO%20e-nav%20CG.PDF
〔6〕 Namje Park, Jin Kwak, Seungjoo Kim, Dongho Won, and Howon Kim, "WIPI Mobile Platform with Secure Service for Mobile RFID Network Environment," LNCS, Advanced Web and Network Technologies and Applications, 3842, pp.741-48, Jan. 2006.
〔7〕 D. Starobinski, A. Trachtenberg, and S. Agarwal, "Efficient PDA Synchronization," IEEE Transcations on Mobile Computing, 2(1), pp.40-51, Apr. 2003.
〔8〕 Namje Park, "Design and Implementation of Mobile VTS Middleware for Efficient IVEF Service", Journal of KICS, 39C(6), pp.466-475, Jun. 2014.
〔9〕 Namje Park, Jin Kwak, Seungjoo Kim, Dongho Won, Howon Kim, "WIPI Mobile Platform with Secure Service for Mobile RFID Network Environment", Advanced Web and Network Technologies, and Applications, LNCS 3842, pp. 741-748, 2006
〔10〕 Namje Park and Hyo-Chan Bang, "Mobile middleware platform for secure vessel traffic system in IoT service environment," Security and Communication Networks, John Wiley&Sons Ltd, Nov. 2014.
〔11〕 Jeehee Lee, Hyunji Jung, Sangjin Lee, "Forensic Investigation Procedure for Real-time Synchronization Service", Journal of The Korea Institute of Information Security & Cryptology, 22(6), pp.1363-1374, Dec. 2012.
〔12〕 Namje Park, Hongxin Hu, and Qun Jin, "Security and Privacy Mechanisms for Sensor Middleware and Application in Internet of Things (IoT)," International Journal of Distributed Sensor Networks, 2016, Oct. 2015.
〔13〕 Agarwal, S., Starobinski, D., and Trachtenberg, A., "On the scalability of data synchronization protocols for PDAs and mobile devices," IEEE Network, 16(4), pp.22-28, Aug. 2002.
〔14〕 Namje Park and Namhi Kang, "Mutual Authentication Scheme in Secure Internet of Things Technology for Comfortable Lifestyle," Sensors, 16(1), pp.1-16, Dec. 2015.
〔15〕 Uppoor, S., Flouris, M. D., and Bilas, A., "Cloudbased Synchronization of Distributed File System Hierarchies," In proceedings of IEEE International conrerence on Cluster Computing Workshops and Posters (CLUSTER WORKSHOPS2010), pp.1-4, Sep. 2010.
〔16〕 Namje Park, "Detection Experimentation and Validation of Web Applications using Both Static and Dynamic Analysis," Information, International Information Institute (Tokyo), 18,(5A), pp.1735-1742, May. 2015

〔17〕 Ho Min Jung, Young Woong Ko, Jae Min Park, Jin San Kong, "A File Synchronization System using Similarity-based Deduplication", Journal of KIISE : Computing Practices and Letters, 18(7), pp.548-552, Jul. 2012.
〔18〕 Namje Park, "Performance Analysis and Improvement for Data Exchange Protocol in Vessel Traffic System," Advanced Science Letters, 21(3), pp.504-507, Mar. 2015.
〔19〕 Merkle, Ralph C. "Security, Authentication, and Public Key Systems." 1979.
〔20〕 Namje Park, "Implementation of Inter-VTS Data Exchange Format Protocol based on Mobile Platform for Next-generation Vessel Traffic Service System," International Information Institute (Tokyo). Information, 17(10A), pp.4847-4856, Oct. 2014.
〔21〕 Namje Park, "Implementation of terminal middleware platform for mobile RFID computing," International Journal of Ad Hoc and Ubiquitous Computing, 8(4), pp.205-219, 2011.
〔22〕 Namje Park, "User Privacy Preserving Mobile RFID Personal Information Security Service System," Journal of Korean Institute of Information Technology, 8(10), pp.87-96, Oct. 2010
〔23〕 Yeonghae Ko, Namje Park, "A Study of IT Centered Smart Grid's STEAM Curriculum and Class for 3rd and 4th Graders in Elementary School," Journal of Korea Association of Information Education, 17(2), pp.167-175, Jun. 2013.
〔24〕 Namje Park, "The Implementation of Open Embedded S/W Platform for Secure Mobile RFID Reader", The Journal of Korea Information and Communications Society, pp.785-793, May. 2010.
〔25〕 Donghyeok Lee, Namje Park, "An Improvement of the eID Online Authentication Scheme for Privacy Enhancement," Journal of Korean Institute of Information Technology, 14(5), pp.89-98, May 2016.

〈저자소개〉

이 동 혁 (Donghyeok Lee) 정회원
2007년 2월: 동국대학교 전자상거래기술전공 공학석사
2007년 6월~2008년 5월:한국전자통신연구원 정보보호연구단 연구원
2008년 11월~2015년 6월:KT 플랫폼개발단 과장
2015년 9월~현재: 제주대학교 컴퓨터교육전공 박사과정
〈관심분야〉 클라우드 보안, 스마트그리드 보안, 데이터베이스 보안

박 남 제 (Namje Park) 종신회원
2008년 2월: 성균관대학교 컴퓨터공학과 박사
2003년 4월~2008년 12월: 한국전자통신연구원 정보보호연구단 선임연구원
2009년 1월~2009년 12월: 미국 UCLA대학교 공과대학 Post-Doc,WINMEC 연구센터 Staff Researcher
2010년 1월~2010년 8월: 미국 아리조나(ASU) 주립대학교 컴퓨터공학과 연구원
2010년 9월~현재: 제주대학교 초등컴퓨터교육전공 교수
〈관심분야〉 융합기술보안, 컴퓨터교육, 스마트그리드, IoT, 해사클라우드 등