Journal of the Korea Convergence Society
Vol. 12. No. 4, pp. 51-60, 2021

ISSN 2233-4890 / eISSN 2713-6353
https://doi.org/10.15207/JKCS.2021.12.4.051

# 체인코드 기반의 파일 무결성 검증 모델

김효종[1], 한군희[2], 신승수[3*]
[1]동명대학교 컴퓨터미디어공학과 학생, [2]백석대학교 컴퓨터공학부 교수, [3]동명대학교 소프트웨어융합보안학과 교수

# Chaincode-based File Integrity Verification Model

Hyo-Jong Kim[1], Kun-Hee Han[2], Seung-Soo Shin[3*]
[1]Student, Dept. of Computers & Media Engineering, Tongmyong University
[2]Professor, Division of Computer Engineering, Baekseok University
[3]Professor, Dept. of Software Convergence Security, Tongmyong University

**요 약** 최근 네트워크 및 하드웨어 기술이 발전하여 블록체인 기술과 보안을 융합하는 연구가 활발히 진행되고 여러 네트워크 기술에 적용되고 있다. 기존의 블록체인을 활용한 기술을 분석하고 제한된 환경에서 비공개 블록체인을 활용한 파일의 무결성을 검증하는 모델을 제안한다. 제안하는 모델은 비공개 블록체인 플랫폼인 하이퍼레저 패브릭의 체인코드로 작성하고 비공개 블록체인 통합 관리 플랫폼인 하이퍼레저 익스플로러를 통해 파일의 무결성 검증을 확인할 수 있다. 제안한 모델의 성능을 개발자 관점과 사용자 관점에서 분석했다. 블록체인 플랫폼을 실행하기 위한 다양한 모듈의 버전에 따른 호환성 문제가 존재하고 체인코드 상태와 그룹 등 제한된 요소만 확인이 가능하다는 결과를 얻었다.

**주제어 :** 블록체인, 체인코드, 사물인터넷, 네트워크, 하이퍼레저

**Abstract** Recent advances in network and hardware technologies have led to active research and multiple network technologies that fuse blockchain technologies with security. We propose a system model that analyzes technologies using existing blockchain and verifies the integrity of files using private blockchain in a limited environment. The proposed model can be written as a chain code of Hyperleisure Fabric, a private blockchain platform, and verified for integrity of files through Hyperleisure Explorer, a private blockchain integrated management platform. The system performance of the proposed model was analyzed from a developer perspective and from a user perspective. As a result of the analysis, there are compatibility problems according to the version of various modules to run the blockchain platform, and only limited elements such as chain code status and groups can be checked.

**Key Words :** Blockchain, Chaincode, IoT, Network, Hyperledger

## 1. 서론

블록체인은 네트워크에서 모든 참여자가 거래에 대한 정보를 검증하고 분산으로 저장하는 기술이다[1]. 신뢰를 기반으로 거래를 진행하여 스마트 계약, 인증 등으로 다양한 서비스에서 이용한다. 대표적으로 금융 및 유통 서비스, 그리고 의료 서비스까지 블록체인을 활용한 분야는 점점 증가하고 있다[2]. 일반적인 블록체인은 공개형 블록체인으로 제약 없이 참여를 원하는 사용자는 모두 참여할 수 있다. 하지만 참여하는 동시에 체인의 형태로

*This work was supported by the BB21+ Project in 2020.
*Corresponding Author : Seung-Soo Shin(shinss@tu.ac.kr)
Received March 11, 2021 Revised March 31, 2021
Accepted April 20, 2021 Published April 28, 2021

해시 정보를 장부에 기록한다. 시간이 지날수록 해시 비용은 증가하고 네트워크의 성능에 영향이 있다[3,4].

대표적인 비공개 블록체인은 리눅스 재단에서 주관하고 오픈소스 프로젝트 중 하나인 하이퍼레저 패브릭이 있다. 하이퍼레저 패브릭은 이더리움의 Solidity와 같이 체인코드라는 언어로 작성하여 블록체인을 구성한다. 작성된 체인코드는 도커 이미지 프로그램에서 실행한다. 도커 이미지는 구축의 시간을 최소화하고 동일한 구축환경을 위해 재단에서 배포하는 파일이다[5].

기존의 하이퍼레저 패브릭을 사용하는 활용 사례는 비즈니스 모델 설계와 증명서 인증, 그리고 저작권 등록 등의 일반적인 모델링 설계 기법이다. 기업이나 제한된 환경, 그리고 일부 특수한 환경에서 작성된 중요한 대외비 파일의 경우, 외부로의 유출로 인해 심각한 문제점이 발생한다[15].

이러한 문제점을 해결하기 위해서 비공개 블록체인을 구성하고 파일의 해시를 기반으로 체인코드를 구성하여 파일의 무결성 검증 모델을 구축하여 외부로 데이터의 유출을 차단하는 모델을 제안한다. 그리고 무결성 검증 방법은 하이퍼레저 패브릭의 스마트 컨트랙트로 사용하는 체인코드를 설계와 구성, 그리고 개발 과정으로 진행한다. 파일의 경우, 해시값을 계산하여 블록체인 네트워크에 사전 등록한다. 사용자가 파일에 대한 검증이 필요할 경우, 블록체인 네트워크에 조회 요청을 하여 파일의 무결성을 확인한다.

본 논문의 구성은 2장에서 블록체인과 스마트계약, 그리고 하이퍼레저 패브릭에 관련된 연구를 분석하고, 3장에서는 체인코드를 활용한 파일 무결성 검증 모델을 제안한다. 4장에서는 제안한 모델을 개발자 관점 및 사용자 관점에서 분석하고, 5장에서 결론을 맺는다.

## 2. 관련 연구

본 장에서는 체인코드를 활용한 파일 무결성 검증 모델에 필요한 블록체인, 스마트 계약과 체인코드, 그리고 하이퍼레저 패브릭에 대하여 알아본다.

### 2.1 블록체인

블록체인은 온라인 서명, 해시함수 등 다양한 기술을 이용하여 비트코인과 이더리움, 그리고 위변조방지 기술 등에 사용되고 공개 블록체인과 비공개 블록체인으로 분류된다. 사용자가 많을수록 블록체인에 참여하여 검증하는 인원도 늘어나게 된다[6]. 블록체인에서 이중지불의 문제점을 비트코인에서는 작업증명으로 합의 시스템을 구현함으로써 해결했다. 여기서 작업증명이란 블록에서 임의 값(nonce)을 찾는 과정으로 일정한 연산 과정이 필요하며 임의 값을 찾을 경우, 블록이 생성된다. 생성된 블록은 참여한 모든 클라이언트에게 알려지게 되고 합의를 한다[7,8].

### 2.2 스마트 계약과 체인코드

스마트 계약은 거래 프로토콜로 1994년 Nick Szabo가 처음 사용한 개념이다. 기존의 서면 계약서를 온라인 계약서 형태로 변화되어 온라인 조건이 만족하면 기존의 계약조건(담보, 지급, 집행, 익명성) 등을 자동으로 수행한다. 그리고 계약조건을 자동으로 수행하기 때문에 중계 비용, 채무불이행, 사기에 의한 손해, 강제 집행비용, 기타 비용이 감소하게 된다[8]. 하이퍼레저 패브릭에서 스마트 계약은 체인코드명으로 사용하고 분산원장의 상태 데이터베이스에 대해 실행되며 거래 제안을 통해 시작된다. 체인코드 실행 및 배포는 채널 네트워크로 송신되어 모든 피어의 분산원장에 적용된다[9,10].

### 2.3 하이퍼레저 패브릭

하이퍼레저 패브릭은 Linux Foundation에서 주관하고 오픈소스 프로젝트인 하이퍼레저 프로젝트에서 개발하고 있는 Private 블록체인이다. 인증 시스템에 의해 인가된 클라이언트만 네트워크에 참여할 수 있고 채널을 통해 원장을 공유, 합의 프로토콜을 사용하여 트랜잭션을 검증을 검증한다[11].

Private 플랫폼 네트워크는 클라이언트, 피어, 그리고 오더러 구성된다. 먼저 클라이언트는 피어에게 트랜잭션을 송신하고 피어는 트랜잭션 과정을 한 뒤 결과를 클라이언트에게 송신한다. 피어로 부터 수신 받은 클라이언트는 데이터를 확인하고 오더링 서비스로 송신한다. 오더러는 클라이언트로부터 수신 받은 트랜잭션을 정렬하여 블록 단위로 생성하고 피어에게 블록을 송신한다. 피어는 블록의 모든 트랜잭션을 검증한다[12,13].

Docker는 닷 클라우드에서 출시한 가상 컨테이너 오픈소스 프로그램이다. 기존 클라우드 컴퓨팅 환경에서 여러 프로그램을 설치 및 설정 등에 많은 시간을 소비한다. 또한 운영환경을 매번 변경해야하는 불편함이 있다. 이러한 불편함을 해결하기 위해 Docker는 서버, 소스코드, 바이너리 등을 컨테이너로 분리하고 운영환경은 변경하지 않는다[14]. 그리고 서비스 환경을 이미지로 생성

및 배포하여 여러 서버에 동일한 환경을 제공한다. 만약 서비스를 업데이트 할 경우, 이미지만 새로 생성하여 배포할 수 있다[15]. Docker의 Diagram은 Fig, 1과 같다.

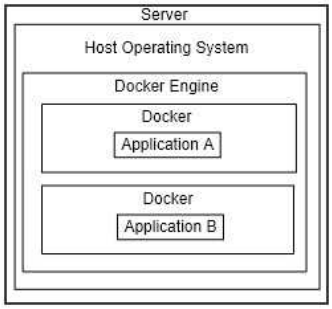


Fig. 1. DApp Environment

## 3. 체인코드 기반의 검증 모델

기존의 구축 방법은 하이퍼레저와 도커 이미지, 그리고 컴포즈를 통해 플레이그라운즈에 접속하여 다양한 비즈니스 모델링을 설정한다. 새로운 모델링을 추가하고 편집하기 위해 항상 관리자가 웹을 통해 로그인하고 관리해야한다. 이러한 기존 방식은 기업 내 특정 파일의 공유가 발생할 때, 매번 관리자를 통해 인증하고 조회를 해야 하는 번거로움과 불필요한 시간적 소모가 생긴다.

이러한 문제점을 해결하기 위해 체인코드 기반의 검증 모델을 제안한다. 본 장에서는 체인코드 기술을 이용한 파일 무결성 검증 모델을 하이퍼레저 패브릭과 하이퍼레저 익스플로러로 분류하여 모델을 구축한다.

### 3.1 시스템 구성

파일 무결성 검증 구성은 노드 그룹, 하이퍼레저 패브릭 플랫폼, 피어, 하이퍼레저 익스플로러로 구성된다. 개발자는 체인코드를 개발하여 클라이언트 애플리케이션으로 실행한 후, 보증인 피어로 송신한다. 보증인 피어는 체인코드를 확인한 후, 채널을 통해 전체 피어에게 배포를 한다. 채널에 연결된 오더러 노드는 트랜잭션을 수신하여 블록체인을 구성한다. 오더러 노드는 트랜잭션을 블록에 적용하고 하이퍼레저 패브릭 네트워크의 피어 노드에게 송신한다. 피어 노드는 다른 피어에게 브로드캐스트 요청을 송신한다. 요청을 수신 받은 피어들은 블록 업데이트를 진행하고 원장을 동기화한다. 이와 같은 과정은 Fig. 2와 같다.

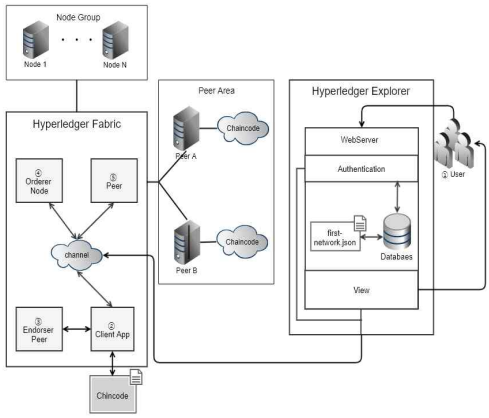


Fig. 2. Network Diagram

하이퍼레저 익스플로러는 클라이언트가 익스플로러에 계정접속 요청을 하면 인증을 통해 계정승인을 하고 인가된 사용자로 권한을 할당한다. 클라이언트는 권한이 할당된 계정을 통해 채널에 대한 블록 요청을 송신한다. 블록에 대한 요청을 수신 받은 피어는 노드와 채널, 그리고 익스플로러를 통해 클라이언트가 확인할 수 있다.

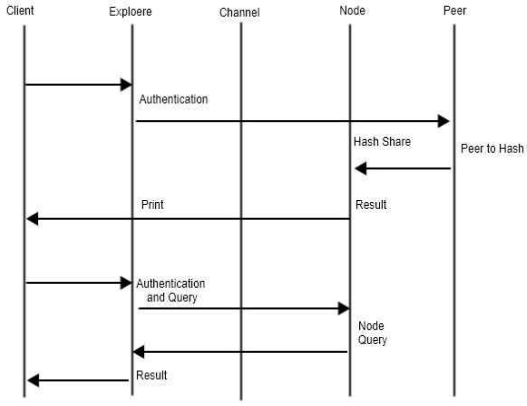


Fig. 3. Flow Chart

### 3.2 시스템 모델

하이퍼레저 패브릭은 체인코드 기술을 이용하여 X.509 Certificate의 무결성을 블록체인으로 검증하고 설계한다. 검증 프로세스를 구현하기 위해서 OS는 Microsoft의 Windows 10 Pro 20H2, Environment는 Oracle의 VirtualBox, Platform은 Hyperlger와 Docker, Dev language는 Google의 Go language를 사용한다. 그리고 개발 의존성에 의해 필요한 Node JS, Git을 사용한다. 이와 같은 체인코드의 개발환경은 Table 1과 같다.

Table 1. Development Environment

| | |
|---|---|
| Operation System | Windows 10 Pro 20H2, Ubuntu 20.04.1 LTS |
| Environment | VirtualBox & Host |
| Platform | Hyperlger, Docker |
| Dev language | Go language |
| Other | Node JS, Git |

### 3.2.1 플랫폼 구축

하이퍼레저 패브릭을 구축하기 전에 의존성을 위해 필요한 가상화 플랫폼, 다양한 프로그램들을 사전 설치한다. 프로그램 패키지인 apt-transport- https는 암호화 통신을 위한 패키지로 ca-certificates, curl, gnupg-agent, software-properties-common과 같이 설치한다.그리고, 하이퍼레저 패브릭 SDK에서 node.js를 사용하므로 필수 요소로 설치한다. 그리고 하이퍼레저 패브릭에서는 체인코드 개발을 위해 Go lang, Java, Javascript 등을 지원한다.

본 논문에서는 Go 언어를 기반으로 작성하기 때문에 Go lang을 설치한다. 또한 하이퍼레저 패브릭을 Docker의 가상화 컨테이너에서 구동해야 하고 피어, 오더러 등의 여러 가상화 컨테이너를 구동해야 하므로 Docker Compose도 필수이다. 플랫폼 설치 구성은 Ubuntu Mirror Server와 Docker Mirror Server 그리고 Go lang Mirror Server 순서로 설치한다. 플랫폼 환경 구성은 Fig. 4와 같다.

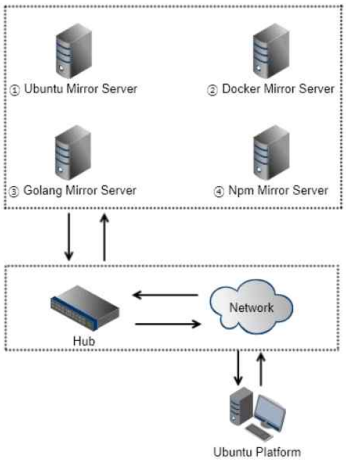


Fig. 4. Platform Environment

기본 플랫폼 구축을 위해 Ubuntu Mirror Server, Docker Mirror Server, Go lang Mirror Server, Npm Mirror Server로 5단계의 Mirror Server로 구성된다. Mirror Server는 네트워크 노드의 복제본으로 다양한 프로토콜로 접근하여 네트워크 및 파일공유 서비스를 지원한다. 첫 번째로 다양한 프로그램과의 호환성을 위해 우분투를 업데이트한다. 기존의 설치된 프로그램을 자동으로 색인하여 업데이트를 진행한다. 두 번째로 우분투 서버의 가상화를 위해 도커 프로그램을 설치하고 GPG Key 인증 절차를 마무리한다. 세 번째로 하이퍼레저 패브릭의 체인코드 개발을 위해 Go 언어를 다운로드 후 설치를 진행 한다. 마지막으로 하이퍼레저 패브릭의 환경 구성을 위해 node js의 패키지 관리자를 설치한다.

① Ubuntu Mirror Server

Ubuntu Mirror Server를 통해 현재 Ubuntu Platform을 최신 버전으로 업데이트 및 업그레이드를 하고 필수 프로그램을 다운로드 후 설치한다.

```
$apt update && apt upgrade
$apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common nodejs
```

② Docker Mirror Server

APT 패키지에 대한 키관리 프로그램으로 다운로드를 검증하기 위해 GPG Key 인증절차를 완료합니다. 인증 결과 “OK” 일 경우, docker repository 등록을 진행한다.

```
$curl -fsSL
https://download.docker.com/linux/ubuntu/gpg|sudo apt-key add -add-apt-repository \
"deb [arch=amd64]
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable“
```

apt-get update 명령어를 통해 기존의 어플리케이션을 업데이트하고 sudo apt-get install 명령어로 도커의 필수 구성 파일을 다운로드 후 설치를 한다. 그리고 systemctl 명령어로 설치된 도커 프로세스의 서비스를 관리자 권한으로 실행한다. 명령어는 다음과 같다.

```
$ apt-get update && sudo apt-get install docker-ce docker-ce-cli containerd.io
$ sudo usermod -a -G docker $USER
```

$ sudo systemctl start docker

$ sudo systemctl enable docker

Hyperledger fabric에서는 다중 컨테이너를 사용하고 관리하기 때문에 자동으로 어플리케이션을 관리하고 공유할 수 있는 Docker Compose 도구를 다운로드 후 설치한다. 그리고 Docker Compose에서는 YAML 파일로 서비스를 정의하고 짧은 명령어로 다중 컨테이너를 실행하거나 종료할 수 있다. Docker Compose 설치 명령어는 다음과 같다.

$ curl-L "https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname -s)-$(uname -m)" -o/usr/local/bin/ docker-compose

Docker compose를 설치를 성공적으로 완료했을 경우, compose의 파일에 대한 실행 권한을 chmod 명령어로 부여한다.

$ chmod +x/usr/local/bin/docker-compose

③ Golang Mirror Server

하이퍼레저 패브릭 체인코드 개발을 위해 구글의 Go lang 프로그래밍 개발 언어를 추가로 설치한다. 설치 명령어는 다음과 같다.

$ wget https://golang.org/dl/go1.15.5.linux-amd64.tar.gz

$ tar-xzvf go1.15.5.linux-amd64.tar.gz

$ sudo mv go//usr/local

Go lang 프로그래밍 개발 언어 설치를 성공적으로 완료했을 경우, 운영체제 전역에서 호출할 수 있도록 환경변수 등록 작업을 한다. 환경변수는 bashrc 파일 검색하여 해당 파일의 마지막에 다음 명령어를 추가한다.

export GOPATH=/usr/local/go

export PATH=$PATH:$GOPATH/bin

④ Npm Mirror Server

하이퍼레저 패브릭의 SDK는 nodejs와 javascript 실행 가능 환경에서 사용하도록 설계되었기 때문에 nodejs의 패키지 관리자로 npm(node package manager)을 설치한다. 설치 명령어는 다음과 같다.

$ curl-L https://npmjs.org/install.sh

### 3.2.2 하이퍼레저 패브릭 구축

하이퍼레저 패브릭 구축은 하이퍼레저 패브릭 Mirror Sever, 하이퍼레저 패브릭 Channel, 하이퍼레저 Explorer, Chaincode로 구성된다. 먼저 우분투 플랫폼에서 하이퍼레저 패브릭 Mirror Sever를 통해 하이퍼레저 패브릭의 이미지를 다운로드 후, Docker 프로그램을 통해 실행한다. Docker는 각각의 가상화 이미지를 Docker Compose 프로그램으로 관리한다. 이후 하이퍼레저 패브릭의 내부 구성은 Orderer Node, Peer, Channel, Client App, Endorser Peer이고 Channel로 통신한다. Channel은 Hyperledger Explorer의 View를 통해 사용자가 블록체인의 상태를 확인한다. 체인코드는 개발자가 설계에 따라서 작성하는 블록체인의 부분이다. 플랫폼 구축은 Fig. 5와 같다

.

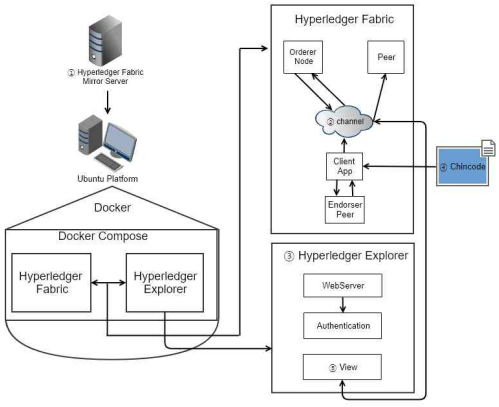


Fig. 5. Hyperledger Fabric

하이퍼레저 패브릭 환경 구축을 위해 Mirror Server, Channel 생성, 하이퍼레저 Explorer, 체인코드 개발, View로 전체 5단계로 구축된다. 첫 번째로 Mirror Server는 외부의 하이퍼레저 패브릭 이미지를 다운로드 후, Docker를 통해 실행한다. 두 번째로 Docker를 통해 블록체인 채널을 생성하고 스크립트 파일을 생성한다. 세 번째로 웹 애플리케이션 연동을 위한 하이퍼레저 Explorer를 설치한다. 네 번째로 개발자는 설계에 따라 체인코드언어로 작성하고 배포를 진행한다. 마지막으로 Docker Compose를 통해 하이퍼레저 패브릭을 관리하고 사용자 관점에서 블록체인을 View로 확인한다.

① 하이퍼레저 패브릭 Mirror Server

하이퍼레저 패브릭 Mirror Server의 파일 주소를 통해 하이퍼레저 패브릭 2.2 버전을 다운로드 후 설치를 진행한다. 만약 다른 버전을 설치할 경우, bash-s x.y.z 명령어로 버전별로 설치할 수 있다. 명령어는 다음과 같다.

$curl-sSL https://bit.ly/2ysbOFE | bash-s 2.2.0

② Channel 생성

하이퍼레저 패브릭 2.2의 네트워크 자동 설정을 위해 test-network 폴더에서 스크립트 파일(network.sh)파일을 "up createChannel" 명령어로 채널을 생성하고 실행한다. 명령어는 다음과 같다.

$./network.sh up createChannel

Docker의 명령어 중에서 "ps-a" 명령어로 현재 도커에서 실행중인 프로세스와 컨테이너를 확인할 수 있다. 만약 컨테이너가 동작하지 않는다면 스크립트파일(network.sh)를 "down" 명령어로 실행 중지하고 다시 "up" 명령어로 컨테이너를 실행한다. 명령어는 다음과 같다.

$docker ps-a

③ 하이퍼레저 Explorer

하이퍼레저 Explorer는 블록과 네트워크 정보, 그리고 체인코드 등의 다양한 목록을 호출하고 배포, 실행하는 웹 애플리케이션 프로그램으로 하이퍼레저 패브릭과 연동하여 블록의 흐름과 노드를 간편하게 웹으로 모니터링 할 수 있다. 하이퍼레저 Explorer 설치를 위한 명령어는 다음과 같다.

$ wget
https://raw.githubusercontent.com/hyperledger/blockchain-explorer/master/examples/net1/config.json

$ wget
https://raw.githubusercontent.com/hyperledger/blockchain-explorer/master/examples/net1/connection-profile/first-network.json-P connection-profile

$ wget
https://raw.githubusercontent.com/hyperledger/blockchain-explorer/master/docker-compose.yaml

하이퍼레저 Explorer를 설치하기 전, 하이퍼레저 패브릭에서 스크립트파일 실행과정에서 생성된 인증서 폴더를 하이퍼레저 Explorer와 연동을 위해 explorer 폴더로 복사를 진행한다.

$cp-r~/fabric-samples/test-network/organizations / ~/explorer

"gedit" 명령어를 통해 docker-compose.yaml파일을 실행하여 볼륨의 설정을 변경한다.

$ gedit docker-compose.yaml
before
volumes:
- ./examples/net1/config.json:/opt/explorer/app/platform/fabric/config.json
- ./examples/net1/connection-profile:/opt/explorer/app/platform/fabric/connection-profile
- ./examples/net1/crypto:/tmp/crypto
- walletstore:/opt/explorer/wallet
after
volumes:
- ./config.json:/opt/explorer/app/platform/fabric/config.json
- ./connection-profile:/opt/explorer/app/platform/fabric/connection-profile
- ./organizations:/tmp/crypto
- walletstore:/opt/wallet

2017년도에 OWASP TOP 10의 섹션 중에서 A2 (Broken Authentication) 섹션으로 취약한 인증에 대한 보안으로 대부분 기본설정을 변경하지 않아 발생되는 취약점이다. 본 논문에서 구축되는 서버는 해당 Broken Authentication 취약점을 인식하여 하이퍼레저 Explorer의 보안 설정을 한다.

"gedit connection-profile/first-network.json" 명령어를 사용하여 json 언어로 작성된 first- network의 설정정보를 편집한다. 그 후 "id", "password" 부분을 관리자 아이디, 비밀번호로 변경한다. 명령어는 다음과 같다.

$ gedit connection-profile/first-network.json
# Default username and password
"id": "exploreradmin",
"password": "exploreradminpw"

체인코드는 Golang, node.js 또는 Java로 작성된 프로그램으로 하이퍼레저 패브릭에서 스마트 컨트랙트를 구현하고 개발한다. Docker 컨테이너에서 단독으로 실행되고 원장을 관리하며 권한이 주어지면 체인코드가 다른 체인코드를 호출하여 접근할 수 있다. 먼저 체인코드 환경변수를 등록하여 호출이 가능하도록 구성을 변경한다.

```
# peer
export PATH=${PWD}/../bin:${PWD}:$PATH
# core.yaml
export FABRIC_CFG_PATH=$PWD/../config/
export CORE_PEER_TLS_ENABLED=true
export CORE_PEER_LOCALMSPID
="Org1MSP"
export
CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
export
CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
export CORE_PEER_ADDRESS=localhost:7051
```

④ 체인코드 개발

체인코드 개발을 위해서 개발자는 프로그램 설계에 따라 변수 A에 args[0]을 저장하고, 변수 Value에 args[1]을 저장한다. 그리고 변수 Length는 파라미터의 길이를 저장한다. 만약 파라미터의 길이가 2가 아니면 Incorrect 문자열을 출력하고 Value값이 NULL이면 Error를 리턴 한다. Value의 값이 정상이면 해시를 조회한 후 리턴 한다. 그리고 PutHash 함수의 파라미터로 전달한다. Input 변수는 PutHash 함수의 결과를 저장하고 만약 PutHash의 함수 결과가 NULL일 경우, Error를 출력하고 아닐 경우 Success를 리턴 한다. 체인코드의 의사코드는 다음과 같다.

```
A ← args[0]
Value ← args[1]
Input ← error
Length ← args
if not Length = 2:
 ..................................... RETURN Incorrect;
if Value = NULL:
 ..................................... RETURN Error;
RETURN Value
Input = PutHash(A,Value)
if not Input = NULL:
 ..................................... RETURN Error;
RETURN Success;
```

⑤ View

```
./network.sh deployCC
```

하이퍼레저 Explorer의 기본설정을 한 뒤, “docker- compose up” 명령어로 도커 컴포즈를 실행한다. 만약 도커의 구성을 편집했거나 체인코드를 변경했을 경우 “docker-compose down” 명령어로 도커 컴포즈를 중지한 후 재시작 한다.

```
$ docker-compose up-d
$ docker-compose down-v
```

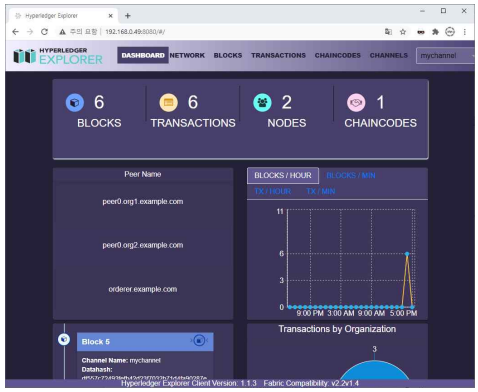


Fig. 6. Chaincode View

## 4. 시스템 모델 분석

시스템 모델 분석은 체인코드 개발자 관점과 체인코드 사용자 관점으로 분류하여 동작 과정을 분석한다. 체인코드를 개발하고 보증인 피어를 통해 인증 받고 오더러 노드로 배포를 진행한다. 개발된 체인코드는 파일의 해시값을 등록하고 조회한다. 체인코드 등록 및 조회는 인가된 사용자만 가능하고 외부의 연결은 통제된다. 인가된 사용자는 하이퍼레저 패브릭 익스플로러 플랫폼으로부터 인증된 파일 해시 정보 여부를 조회한다.

### 4.1 체인코드의 개발 분석

기존의 방식은 하이퍼레저와 도커 이미지, 그리고 컴포즈를 통해 비즈니스 모델링을 설정한다. 새로운 모델링을 추가하고 편집하기 위해 로그인하고 주기적으로 관리한다. 이러한 기존 방식은 매번 인증하고 조회를 해야 하는 불필요한 시간적 소모가 생긴다.

제안한 검증 모델에서 체인코드의 개발 분석은 체인코드, 클라이언트 애플리케이션, 보증인 피어, 피어노드,

오더러 노드로 5단계로 구성된다. 보증인 피어는 클라이언트 애플리케이션에서 트랜잭션 요청을 수신하면 트랜잭션을 검증한다. 검증내용은 요청된 세부 정보와 요청자의 권한을 확인한다. 그리고 체인코드를 실행하고 결과를 사전 테스트한다. 사전 테스트 단계에서는 분산원장과 블록을 업데이트하지 않는다. 사전 테스트 결과가 정상적인 방법을 통해 거래가 발생하지 않을 경우 해당 거래를 차단한다.

오더러 피어는 하이퍼레저 패브릭 네트워크의 핵심으로 블록체인 네트워크 전체에서 동일한 분산원장 상태를 유지한다. 그리고 권한에 따라 블록을 생성하고 브로드캐스트를 통해 피어에게 송신한다. 피어 노드는 분산원장 업데이트를 수신하고 다른 개별 피어에게 브로드 캐스트로 원장 업데이트를 진행한다. 이와 같은 과정은 Fig. 7과 같다.

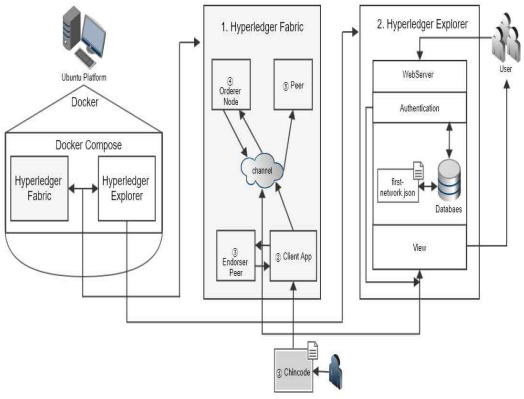


Fig. 7. Chaincode Development Analysis

체인코드 개발 분석과정에서 첫 번째로 하이퍼레저 패브릭의 개발자 관점에서 동작원리에 대해 분석한다. 블록체인 개발자는 프로그램 설계에 따라 체인코드를 작성하여 클라이언트 애플리케이션으로 배포를 요청한다. 그 후 클라이언트 애플리케이션은 보증인 피어로 수신 받은 체인코드를 검증하고 트랜잭션을 진행한다.

트랜잭션을 수신 받은 오더러 노드는 블록체인을 구성하고 피어노드에게 송신한다. 피어노드는 브로드캐스트로 모든 노드에게 최신 블록으로 업데이트를 진행하고 분산원장을 동기화한다.

① 블록체인 개발자는 클라이언트 애플리케이션으로 체인코드 배포요청을 송신한다.
② 체인코드를 수신 받은 클라이언트 애플리케이션은 보증인 피어로 전체에게 배포하고 클라이언트 애플리케이션으로부터 요청에 대한 정보를 확인하여 트랜잭션 한다. 그리고 체인코드를 실행하고 클라이언트 애플리케이션으로 응답을 송신한다.
③ 보증인 피어는 트랜잭션 상태에 대한 성공이나 실패를 반환한다.
④ 클라이언트 애플리케이션은 성공한 트랜잭션을 오더러 노드로 송신하여 블록체인을 구성한다. 오더러 노드는 트랜잭션을 블록에 적용하고 하이퍼레저 패브릭 네트워크의 피어 노드에게 송신한다.
⑤ 수신 받은 피어 노드는 다른 피어에게 브로드캐스트 요청을 송신한다. 브로드 캐스트 요청을 수신 받은 피어들은 최신 블록으로 업데이트를 진행하고 분산 원장을 동기화한다.

### 4.2 체인코드의 사용자 분석

하이퍼레저 패브릭의 사용자 관점에서 동작원리에 대해 분석한다. 사용자는 블록체인 업무를 위해 하이퍼레저 익스플로러로 구성된 웹 서버로 접속한다. 그 후 사용자는 로그인 정보를 입력 후 검증시간을 대기한다. 검증시간은 네트워크 상태에 따라 매번 달라질 수 있다. 검증이 끝난 인가된 사용자는 하이퍼레저 익스플로러 뷰로 블록체인의 정보를 확인한다.

① 하이퍼레저 익스플로러로 구성된 웹 서버로 사용자는 접속을 요청한다.
② 접속 요청을 수신 받은 하이퍼레저 익스플로러는 설정 정보와 데이터베이스 정보를 조회하여 인가된 사용자 여부를 확인한다.
③ 인가된 사용자일 경우, 체인코드 네트워크망에 다이렉트 요청은 불가하며 하이퍼레저 익스플로러를 통한 대리인으로 피어 노드에게 정보를 요청을 송신한다.
④ 수신 받은 피어 노드는 블록 정보를 하이퍼레저 익스플로러 뷰로 송신하여 사용자가 확인할 수 있다.

## 5. 결론

하이퍼레저 패브릭와 익스플로러는 기업형 블록체인으로 금융, 기업 등 다양한 환경에서 활용된다. Docker의 가상화 기술을 융합하고 하이퍼레저 패브릭에서 체인코드를 기반으로 파일 무결성 검증 모델을 제안한다. 체인코드 기반의 무결성 검증 모델은 체인코드 개발자가 블록을 추가 및 조회할 수 있도록 소스코드를 개발하고

클라이언트 앱으로 배포를 한다.

클라이언트 앱은 보증인 피어로 체인코드를 송신하고 보증인 피어에서 체인코드를 사전 실행하여 체인코드 여부를 검증한다. 그리고 보증인 피어의 검증 단계에서 체인코드의 문제점이 발생할 경우, 배포는 중지된다. 모든 검증을 완료한 체인코드는 채널을 통해 모든 노드로 배포된다. 배포된 체인코드에 따라 오더러 노드는 블록을 생성하고 모든 피어에게 송신하여 분산 원장을 생성한다. 이후 관리자는 무결성 검증할 파일의 해시를 블록 단위로 추가하고 검증할 사용자는 체인코드를 통해 블록의 무결성을 검증한다.

향후 파일을 검증 서버로 업로드하여 자동으로 해시값을 블록체인 네트워크에 등록하고 무결성을 검증 하는 모델링에 대해 연구가 필요하다.

## REFERENCES


[1] J. S. Kim, S. C. Kim & N. J. Park. (2021). Blockchain-based Product Carry-in/out Management Mechanism. *The Journal of Korean Institute of Information Technology, 19(1)*, 131-138.

[2] J. H. Kim. (2020). Fault Tolerance Enhancement for Distributed Storage Blockchain Systems. *Journal of the Korea Institute of Information and Communication Engineering*, 1558-1561.

[3] S. H. Lee & K. C. Kim. (2021). Proposal of Blockchain Structure with Real-Time Capability Suitable for 5G-Based Service Provision. *The Journal of Korean Institute of Communications and Information Sciences, 46(1)*, 102-113.

[4] M. Y. Kim, H. S. Lee & J. D. Kim. (2020). A Blockchain Copyright Information Registration System for Content Protection of Online Sharing Platforms. *Journal of the Korea Institute of Information and Communication Engineering, 24(12)*, 1718-1721.

[5] J. K. Lee. (2018). A Docker Container Case Study for Implementing Block Chain Distributed General Ledger. *Korean Computers and Accounting Review, 16(1)*, 27-41.

[6] J. C. Yeom, Y. S. Lee & In-Ho Ra. (2019). Proposal of PublicKey_Based Node Authentication Method for Secure Private Block Chain Environment. *Korea Institute Of Communication Sciences*, 466-467.

[7] W. S. Park, D. Y. Hwang & K. H. Kim. (2018). A Study on Certificate Issuance System Using Private Block chain. *Korea Institute Of Communication Sciences*, 1192-1193.

[8] B. J. Cho. (2019). BOTP(Blockchain One Time Password) System. *The Korean Institute of Information Scientists and Engineers*, 1221-1223.

[9] H. G. Kim, S. H. Jung. (2020). IoT Authentication System Using Blockchain and TOTP. *Journal of the Korea Society of Computer and Information, 25(2)*, 113-122.

[10] G. Y. Jung. (2018). A Study on the legal nature of cryptocurrency and a smart contract. *Korea commercial law association*, 109-150.

[11] H. K. Kwon, H. J. Kim, K. B. Jang & H. J. Seo. (2020). An Improvement of missing pet rescue system based on Smart Contract. *Journal of the Korea Institute of Information and Communication Engineering, 24(6)*, 786-794.

[12] J. Y. Kim, J. H. Kwon, S. I. Jang, Y. K. Kim, M. J. Son & M. H. Kim. (2020). A Method to reduce transaction endorsement time based on network latency in Hyperledger Fabric. *Proceedings of the Korean Society of Computer Information Conference, 28(2)*, 41-44.

[13] S. I. Jang, J. H. Kwon, J. Y. Kim, C. H. Im & M. H. Kim. (2020). Study on Chaincode execution cost base endorsing peer routing method to reduce endorsement time in Hyperledger Fabric. *Proceedings of the Korean Society of Computer Information Conference, 28(2)*, 37-40.

[14] K. H. Lee. (2018). A Study on the type of BlockChain(Ethereum, Hyperledger Fabric). *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, 442-443.

[15] J. K. Lee & H. Y. Choi. (2018). A Case Study of Using Hyperledger Composer based on Docker Container for Implementing Block Chain Distributed Ledger. *Korean Computers and Accounting Review, 16(2)*, 1-18.



김 효 종(Hyo-Jong Kim) [정회원]

· 2016년 2월 : 동명대학교 정보보호학과(공학사)
· 2017년 2월 ~ 현재 : 동명대학교 컴퓨터미디어공학과 석사과정
· 관심분야 : 웹 크롤링, 빅데이터 분석, 네트워크 보안
· E-Mail : khj47561404@gmail.com

한 군 희(Kun-Hee Han) [종신회원]

· 2001년 3월 ~ 현재 : 백석대학교 컴퓨터공학부 교수
· 관심분야 : 멀티미디어, 유비쿼터스, DB보안, 암호 프로토콜/알고리즘
· E-Mail : hankh@bu.ac.kr

신 승 수(Seung-Soo Shin) [정회원]

· 2001년 2월 : 충북대학교 수학과 (이학박사)
· 2004년 8월 : 충북대학교 컴퓨터공학과(공학박사)
· 2005년 3월 ~ 현재 : 동명대학교 SW융합보안학과 교수
· 관심분야 : 암호프로토콜, 네트워크 보안, U-헬스 케어, IoT, 데이터분석
· E-Mail : shinss@tu.ac.kr