KIPS Trans. Comp. and Comm. Sys.
Vol.11, No.5 pp.147~156
ISSN: 2287-5891 (Print), ISSN 2734-049X (Online)

https://doi.org/10.3745/KTCCS.2022.11.5.147

A Study on the Blockchain-Based Access Control Using Random-List in Industrial Control System

Kang Myung Joe† · Kim Mi Hui††

ABSTRACT

Industrial control systems that manage and maintain various industries were mainly operated in closed environment without external connection, but with the recent development of the Internet and the introduction of ICT technology, the access to the industrial control system of external or attackers has become easier. Such incorrect approaches or attacks can undermine the availability, a major attribute of the industrial control system, and violation of availability can cause great damage. In this paper, when issuing commands in an industrial control system, a verification group is formed using a random list to verify and execute commands, and a trust score technique is introduced that applies feedback to the verification group that conducted verification using the command execution result. This technique can reduce overhead generated by random generation in the process of requesting command verification, give flexibility to the verification process, and ensure system availability. For the performance analysis of the system, we measured the time and gas usage when deploying a smart contract, gas usage when verifying a command. As a result, we confirmed that although the proposed system generates a random list compared to the legacy system, there was little difference in the time when it took to deploy smart contract and that the gas used to deploy smart contract increased by about 1.4 times in the process of generating a random list. However, the proposed system does not perform random operations even though the operation of command verification and confidence score technique is performed together during the command verification process, thus it uses about 9% less gas per verification, which ensures availability in the verification process.

Keywords : Blockchain, Smart Contract, Industrial Control System, Access Control

산업제어시스템에서 랜덤리스트를 이용한 블록체인 기반 접근제어 방식에 관한 연구

강 명 조† · 김 미 희††

요 약

다양한 산업을 관리하고 유지하는 산업제어시스템은 주로 외부와의 연결 없이 폐쇄적으로 운영됐지만 최근 인터넷의 발전과 ICT 기술의 도입으로 외부나 공격자의 산업제어시스템에 접근이 쉬워졌다. 잘못된 접근이나 공격은 산업제어시스템의 주요 속성인 가용성을 해칠 수 있으며, 가용성이 침해될 경우 큰 피해가 발생할 수 있다. 본 논문에서는 산업제어시스템에서 명령을 내릴 때 랜덤리스트를 생성해 검증그룹을 구성하여 명령을 검증 후 실행하며, 명령 실행 결과를 이용해 검증을 진행한 검증그룹에 피드백을 적용하는 신뢰 점수 기법을 도입한다. 이를 통해 명령 검증 요청과정에서 랜덤 생성에 발생하는 오버헤드를 줄일 수 있으며, 검증 과정에 유연성을 부여하고 시스템의 가용성을 보장할 수 있다. 시스템의 성능 분석을 위해 스마트 계약 배포 시 걸리는 시간과 가스 사용량, 명령 검증 시 가스 사용량을 측정했다. 그 결과, 기존시스템과 비교해 랜덤리스트를 생성하지만, 스마트 계약 배포에 걸리는 시간은 거의 차이가 없음을 확인했고 스마트 계약 배포에 사용되는 가스는 랜덤리스트 생성과정에서 약 1.4배 증가함을 확인했다. 하지만, 명령 검증 과정에서 명령 검증과 신뢰 점수 기법의 연산을 함께 진행함에도 랜덤 연산을 하지 않아 검증 1회당 약 9% 적은 가스를 사용해 검증 과정에 가용성을 보장한다.

키워드 : 블록체인, 스마트 계약, 산업제어시스템, 접근제어

1. 서 론

최근 산업제어시스템(Industrial Control System: ICS)의 규모가 확대되고 ICT(Information & Communication Technology)와 산업제어시스템이 결합한 스마트 팩토리가 활성화되며 공격자가 직접 접근할 수 있는 인터넷을 통한 보안 위협이 증가하고 있다[1,2]. 산업제어시스템을 대상으로 한 공격으로는, 호주에서 발생한 하수처리시스템 제어권 탈취사고, 미국에서 발생한 Davis-Besse 원자력발전소 사고. 영국 내 최대 자동차 생산공장인 선덜랜드 자동차 랜섬웨어 감염(2017), 일본 혼다자동차 생산시설 랜섬웨어 감염(2017), 대만 TSMC 반도체 공장 워너크라이 감염(2018), 노르웨이

※ 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2018R1A2B6009620).
† 비 회 원 : 한경대학교 컴퓨터응용수학부 석사과정
†† 종신회원 : 한경대학교 컴퓨터응용수학부 컴퓨터시스템연구소 교수
Manuscript Received : November 23, 2021
First Revision : January 24, 2022
Accepted : March 2, 2022
* Corresponding Author : Kim Mi Hui(mhkim@hknu.ac.kr)

노르스크 하이드로 알루미늄 생산업체, 미국 화학업체 헥시온 및 모멘티브 생산시설 LockerGoga 랜섬웨어 감염(2019) 등이 있으며 이러한 외부의 공격으로 발생하는 문제는 엄청난 규모의 경제적 손실과 사회적 손실이 발생할 수 있어 외부로부터 적절한 접근제어를 수행해야 한다[3,4]. 최근 산업제어시스템에 대한 국가별 사이버 보안 현황을 살펴보면, 미국은 국립표준기술연구소(National Institute of Standards and Technology: NIST)에서 산업제어시스템의 보안을 위한 규정들을 정의하고 있으며, 사이버 보안 및 인프라 보호기관(Cybersecurity and Infrastructure Security Agency: CISA)에서 산업제어시스템 보안 평가 도구로서 CSET(Cyber Security Evaluation Tool)를 제공한다. 일본은 JPCERT/CC (Japan Computer Emergency Response Team Coordination Center)에서 산업제어시스템을 평가할 수 있는 항목을 제공하는 SSAT(SCADA Self Assessment Tool)를 개발하여 제공하고 있다. 국내에서는 2017년에 한국전자통신연구원의 부설 연구소인 국가보안기술연구소에서 산업제어시스템 보안 요구사항을 등록했고 운영계층, 제어계층, 현장장치 계층으로 산업제어시스템을 구성했다[5].

대부분 거대한 규모로 이루어진 산업제어시스템이 공격으로 인해 중앙 서버가 타격을 입으면, 전체 시스템이 마비될 뿐만 아니라 데이터의 복구가 매우 힘들다. 하지만 블록체인을 통해 산업제어시스템을 P2P(Peer-To-Peer) 네트워크로 구축한 뒤 분산 시스템으로 운영한다면, 공격으로 인해 특정 노드가 마비되더라도 다른 노드에 큰 영향을 주지 않을 뿐만 아니라 모든 노드가 같은 장부를 갖는 블록체인의 특성을 이용해 데이터를 보존할 수 있다[6].

기존시스템[7]은 스마트 계약 기반 산업제어시스템 접근제어 메커니즘을 제안했다. 운영계층, 제어계층, 현장장치 계층으로 나누어진 산업제어시스템에서 운영계층으로부터 명령을 받은 제어계층이 현장장치 계층으로 명령을 전달하기 전에 운영계층으로부터 명령을 검증받는 접근제어 메커니즘이다. 기존시스템의 제어계층은 명령에 대한 검증을 요청할 때, 랜덤을 사용하여 운영계층에게 검증을 요청하는데, 어떤 방식의 랜덤을 사용하는지 명시되어 있지 않다. 랜덤을 생성하는 다양한 방식이 있지만, 고유적인 랜덤 생성 방식이 없는 블록체인의 특성으로 인해 검증 과정의 구현에 따라 스마트 계약의 수행시간이 달라지는 문제점이 있다.

본 논문은 산업제어시스템 대상 기존 스마트 계약 기반 접근제어 메커니즘의 개선하여 가용성을 강화한 접근제어 기법을 제안한다. 제안 방법의 타당성을 검증하기 위해 실험 환경을 구성하고 관련 성능 평가(스마트 계약 배포 시 걸리는 시간과 가스 사용량, 명령 검증 시 가스 사용량 측정)를 수행하였다.

서론에 이어 2장 관련 지식에서는 제안시스템을 이해하기 위한 개념들을 소개한다. 3장 제안시스템에서는 본 논문에서 제안하는 시스템을 소개한다. 4장 성능 평가에서는 기존시스템과 제안시스템의 비교를 위한 실험 방법을 소개하고, 가용성을 중심으로 실험결과를 분석하여 서술한다. 5장 결론에서는 산업제어시스템 보안과 가용성의 의의를 환기하고, 제안시스템과 향후 연구에 관한 생각을 정리한다.

2. 관련 지식

2.1 산업제어시스템

산업제어시스템은 감시 제어 및 자료수집(Supervisory Control And Data Acquisition: SCADA) 시스템, 분산제어시스템(Distributed Control System: DCS), 프로그래머블 로직 컨트롤러(Programmable Logic Controller: PLC)와 같은 기타 제어시스템 구성 등 여러 가지 제어시스템을 포함하는 용어로써 사용된다.

Fig. 1은 산업제어시스템의 일반적인 구조를 보여주고 있다. 일반적으로 운영계층(Operation Layer), 제어계층(Control Layer), 현장계층(Field Layer)으로 이루어져 있으며 외부 인터넷과 내부 인터넷 사이에서 접근제한을 수행하여 시스템을 보호할 수 있도록 하는 DMZ(Demilitarized Zone)를 통해 연결되어 있다. 운영계층은 프로세서 시스템과 시스템 운영자 간의 인터페이스인 HMI(Human Machine Interface), 시스템의 운영 및 유지 보수를 위해 설계된 고성능 컴퓨팅 플랫폼 EWS (Engineering Workstation), EMP(Employee) 등으로 구성되어있고 산업제어시스템을 운영하고 관리하는 직접적인 주체다. 제어계층은 시스템을 제어하고 감시하며 다양한 데이터를 받아들이는 장치인 PLC, 전체 시스템을 여러 개의 시스템으로 나누어 관리해 부하를 줄이는 DCS, 시스템에서 요구되는 데이터를 원격으로 측정하고, 각 단말과 시스템의 인터페이스 역할인 RTU(Remote Terminal Unit) 등으로 구성되어있으며 운영계층이 내린 명령을 실행해 현장장치계층을 제어하고 현장장치계층의 명령 실행 결과를 운영계층에게 전달한다. 현장장치계층은 센서와 액추에이터 등으로 구성되어있고 이들은 시스템의 관리대상과 가장 가까이 있다. 주로 제어계층으로부터 명령을 받아 데이터를 측정하고, 측정한 값을 반환한다.

2.2 블록체인

블록체인[8]이란, 익명의 저자인 Satoshi Nakamoto가

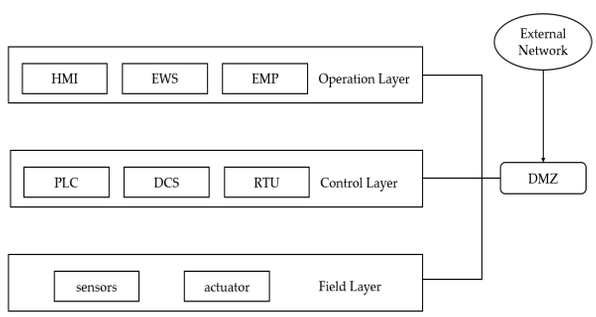

Fig. 1. Structure of Industrial Control System

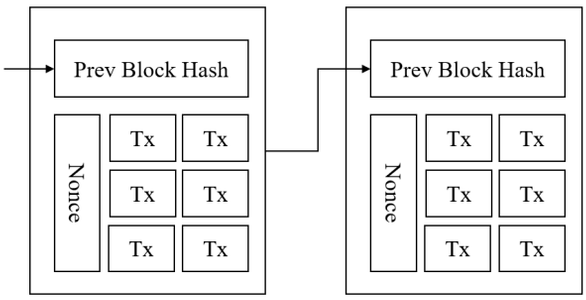

Fig. 2. Structure of Bitcoin Block

2008년 공개한 비트코인 백서에서 제안한 기술로, 중앙 서버를 이용하는 방식이 아닌, P2P 기술을 중심으로 구성된 분산 장부 저장 방식이다.

Fig. 2는 블록체인의 주요 구성요소인 블록을 나타내며, 블록과 트랜잭션으로 구성되어있고 블록 속에는 다수개의 트랜잭션이 포함되어 있다. 트랜잭션이란, 특정 데이터 집합을 트랜잭션 발행자의 개인 키로 서명한 데이터 단위를 뜻하며, 블록체인 네트워크의 종류에 따라 달라진다. 예를 들어 비트코인에서의 트랜잭션은 수신자의 공개키를 이용하여 비트코인 소유 상태를 해쉬한 데이터 집합의 의미로 사용되며, 이더리움[9]에서의 트랜잭션은 스마트 계약을 위한 데이터나, 가스 수수료, 송신자의 전자서명 등으로 이루어진 데이터 집합을 의미한다.

블록체인의 P2P 네트워크에 참여하고 있는 객체를 노드라고 칭하며, 블록체인이 신뢰성과 무결성을 잃지 않도록 네트워크에 참여한 모든 노드가 지켜야 할 규칙을 합의 규칙이라고 칭한다. 합의 규칙은 이전 블록의 해시값을 이용하여 새로운 블록을 연결하는 데 이용한다. 합의 규칙의 종류는 암호화폐마다 다르며, 규칙의 내용도 조금씩 다르다. 대표적인 합의 알고리즘은 PoW (Proof-of-Work), PoS (Proof-of-Stake), PBFT (Practical-Byzantium Fault Tolerance) 등이 있다. PoW는 컴퓨팅 파워를 이용해 해시값을 먼저 계산함에서 경쟁하는 합의 알고리즘이다. PoS 자산의 지분의 양을 통해 합의하는 알고리즘으로, PoW의 에너지 낭비 이슈를 해결하고자 제시되었다. PBFT은 네트워크에 참여한 노드 간 다수결 투표를 이용하는 합의 알고리즘이다. 비트코인에서는 PoW를 사용 중이고 이더리움에서는 PoS로 변환할 예정이다[10].

2.3 스마트 계약

스마트 계약[11]은 Nick Szabo가 1996년에 제안한 기법으로 특정 조건을 만족하거나 미리 정의된 일정에 따라 자동으로 수행되는 전자거래다. 제안한 당시, 계약을 중개할 수 있는 제3의 신용기관이 존재하지 않아 신뢰성을 보장하지 못하고 위변조가 쉬운 문제점으로 인해 주목받지 못했지만, 블록체인 기술이 발전하고 튜링 완전 언어인 솔리디티[12]를 탑재한 이더리움의 등장으로 2000년대에 들어 재조명받고 있다. 현재에도 스마트 계약의 대표적인 개발 플랫폼은 이더리움으로, 암호화폐 부문에서 비트코인보다 늦게 출범했지만, 스마트 계약 분야를 이끌고 있다[13,14]. 최근에는 권한 관리, 금융, 게임, 보험 등의 분야에서 주로 사용된다. 예로 게임 분야에서는 게임 속 내용에 대한 조작 및 사기를 방지하기 위해 스마트 계약을 사용한다. 게임의 참여자들은 게임 개발자나 데이터 제공자의 게임 속 아이템 획득 확률 조작이나 시세 조작으로부터 스마트 계약이 제공하는 신뢰성을 통해 게임을 자체적으로 검증하며 즐길 수 있다.

2.4 사전 연구

조민정, 이창훈은 2019년 스마트 계약 기반의 산업제어시스템 접근제어 방식을 제안했다. 제안시스템은 폐쇄 망으로 구성된 산업제어시스템의 운영계층과 제어계층에서 사용하는 시스템으로 프라이빗 블록체인으로 구성하며, 인적 자원 관리 스마트 계약, 운용 로그 스마트 계약, 검증 스마트 계약 등으로 구성되어있다. 인적 자원 관리 스마트 계약은 직원 정보 관리 시스템과 블록체인을 통합하여 산업제어시스템을 제어할 수 있는 직원 권한에 대한 갱신과 폐기작업을 수행한다. 운용 로그 스마트 계약은 산업제어시스템의 무결성과 보안을 위협하는 행위에 대한 책임 추적성을 확보하기 위한 계약으로, 운영계층으로부터 처리된 행위를 기록하여 무결성과 책임 추적성을 동시에 만족할 수 있도록 한다. 검증 스마트 계약은 운영계층이 제어계층으로 전달한 명령을 확인한 제어계층이 또 다른 운영계층에게 해당 명령 실행에 대한 검증을 요청하고, 요청을 받은 다수의 운영계층이 이를 검증하여 명령을 실행할 수 있도록 하는 계약이다[7].

Fig. 3은 기존시스템[7]의 운영계층과 제어계층의 구조를 나타낸다. 운영계층은 직원과 관리자, HMI, EWS 등으로 이루어지고, 제어계층은 PLC, DCS, RTU 등으로 구성되어있다. 관리자는 직원들의 권한을 갱신하는 역할의 운영계층 개체로, 직원이 갱신 요청 트랜잭션을 통해 권한 갱신 요청을 하면, 관리자가 판단하여 수정/폐지 트랜잭션을 통해 트랜잭

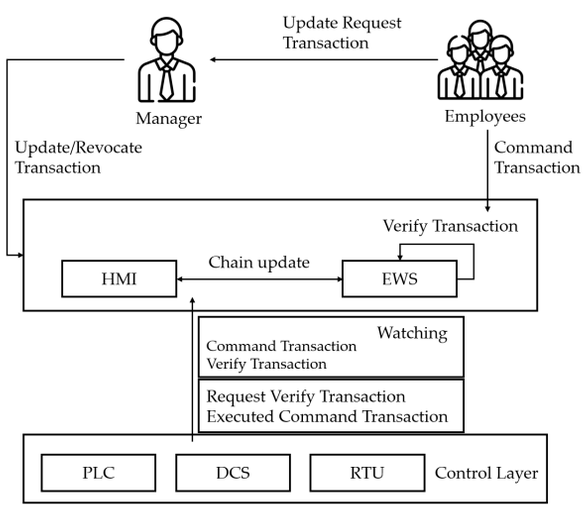

Fig. 3. Structure of Based System

션의 내용에 대한 발행자의 권한과 관련된 내용을 블록체인에 기록한다. 제어계층은 운영계층이 명령 트랜잭션으로 내린 명령을 확인하고 검증요청 트랜잭션을 통해 명령 검증을 요청한다. 검증 트랜잭션을 통해 검증이 완료된 명령은 명령 실행 트랜잭션을 통해 실행된다.

Sultana, Almorgren, Akbar은 2020년 IoT(Internet Of Thing) 기기들을 위한 스마트 계약 기반 데이터 공유 및 접근제어 통합 시스템을 제안했다[15]. 제안시스템은 ACC (Access Control Contract), RC(Register Contract), JC (Judge Contract) 3가지 스마트 계약을 사용하여 효율적이고 능동적인 접근제어를 이룬다. ACC는 전체 시스템의 모든 접근제어를 관리하며, RC는 시스템의 사용자를 인증하고 JC는 사용자의 잘못된 행위를 탐지하여 패널티를 부여한다. 공유 서비스 및 접근제어 관리는 이더리움 네트워크를 사용함으로써 달성했으며, IoT 사용자에 대한 접근 권한을 제공하기 위해 권한 수준을 여러 개로 세분화하여 설정해 각 계층 간 안전한 공유가 이루어지도록 했다.

Albreiki, Alqassem, Salah, Rehman, Svetinovic은 2019년 오라클 서비스와 블록체인을 이용해 IoT 기기가 사용하는 다양한 스토리지 서비스와 저장된 데이터의 유형을 고려한 권한 기반 접근제어 기법을 제안했다[16]. 오라클은 블록체인과 서비스 제공자, 원격 클라이언트 간의 인터페이스 역할을 하는 게이트웨이로, 블록체인 외부에서 스마트 계약이나 블록체인에 데이터를 넘겨주기 위해 사용된다. 제안시스템은 IoT Data Access SC(Smart Contract), Reputation SC, Aggregator SC 3가지 스마트 계약을 사용하여 오라클, IoT 기기, 사용자 간의 상호작용을 이룬다. IoT Data Access SC는 IoT 데이터에 접근하는 사용자의 권한을 검증하고 모든 접근제어를 관리하며 특정 IoT 기기에 대한 사용자의 접근 요청을 받고 사용자와 오라클을 연결하는 Aggregator SC에 접근 요청을 보낸다. Reputation SC는 모든 오라클의 평균 평가 점수를 계산하고 기록한다. 평가 점수는 데이터에 접근하기 위한 사용자의 요청 후 Aggregator SC에게 받는다. Aggregator SC는 오라클 집합에 사용자가 요청한 데이터를 요청하고, 해당하는 데이터의 해시값을 받는다. 그 후, 관련된 오라클에 대한 점수를 Reputation SC에 전송하고, 점수가 높은 오라클을 선택하여 엑세스 토큰을 생성해 사용자와 선택한 오라클에 전송한다. 사용자와 오라클은 엑세스 토큰을 이용하여 요청한 IoT 데이터에 접근할 수 있다.

[15]와 [16] 시스템은 IoT 환경에서 블록체인과 스마트 계약을 이용해 접근제어를 제공한다. 두 시스템 모두 특정 사용자가 목표 IoT 기기에 대해 접근 요청을 할 경우, 사용자의 권한, 신원 등을 인증하고 검증하여 다음 프로세스를 진행한다. 만약 단순한 정보 열람, 명령일 경우에도 이런 검증 과정이 발생한다면, 불필요한 자원 낭비로 인해 가용성이 침해될 수 있다. 본 논문에서는 산업제어시스템의 가용성을 위해 검증이 필요한 명령과 필요하지 않은 명령을 구별해 검증을 진행하므로 불필요한 자원 낭비를 방지할 수 있으며, 검증 과정에 가용성을 추가할 신뢰 점수 메커니즘이 포함되어 있어 검증에 대한 책임 추적성을 부여한다. 또한, 퍼블릭 블록체인 환경에서 구동되는 두 시스템과 달리, 제안시스템은 산업제어시스템의 유형에 따라 퍼블릭, 프라이빗 블록체인 환경으로 자유롭게 구성할 수 있다.

3. 제안시스템

기존시스템은 제어계층의 검증요청을 랜덤으로 구현하는 방법을 구체적으로 정의하지 않았다. 가용성이 중요한 산업제어시스템에서 랜덤과 관련된 연산이 지속해서 발생한다면, 시스템의 자원 낭비가 발생할 수 있다. 본 논문에서는 이와 같은 문제점을 해결하기 위해 랜덤리스트를 도입하여 검증 과정에 필요한 연산의 최소화를 통해 다른 랜덤 구현보다 가용성을 만족할 수 있게 한다. 랜덤리스트를 이용해 구성된 검증그룹은 추후 직원 평가지표로 사용할 신뢰 점수 피드백의 대상이 되며, 신뢰 점수를 기반으로 명령 검증에 대한 책임 추적성 또한 만족할 수 있게 한다.

3.1 랜덤리스트

Fig. 4는 운영계층 직원이 10명이 있는 상황으로 가정한 랜덤리스트를 나타낸다. 명령을 내리기 위한 계약을 배포할 때, 계약의 생성자 함수를 통해 검증그룹 구성을 위한 랜덤리스트를 생성한다. 먼저 직원 수 만큼의 길이를 가진 인덱스 10의 기본 리스트를 만든 뒤, keccak 해시 함수를 통해 난수를 생성한다. 생성한 난수를 기본 리스트의 인덱스와 모듈러 연산을 진행해 10개의 값(Fig. 4. % Value)을 도출한다. 이때 도출된 값을 인덱스가 높은 순서대로 기본 리스트의 인덱스로 사용하여 해당 인덱스에 있는 값을 임시저장소인 Temp에 저장한다. Temp에 값을 넘겨준 공간에는 앞서 난수와 모듈러 연산을 진행한 인덱스에 있는 값을 넣는다. 모듈러 연산을 진행한 인덱스에는 Temp에 저장해놓은 값을 넣는다. 이러한 과정을 통해 제안시스템은 난수를 1번 생성하여 검증 과정에 필요한 랜덤리스트를 만들 수 있다.

블록체인에서 난수 생성은 다양한 방식으로 구현될 수 있다. 대표적으로 블록의 타임스탬프 값을 이용하거나 채굴 난

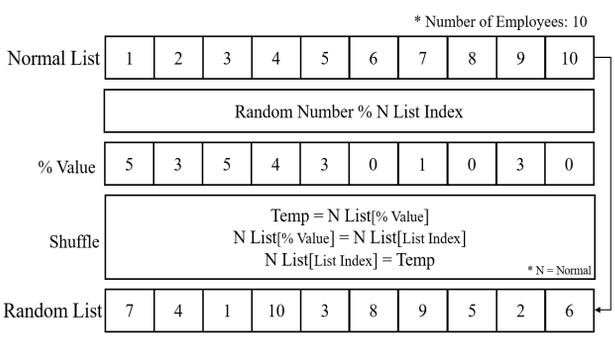

Fig. 4. Random List

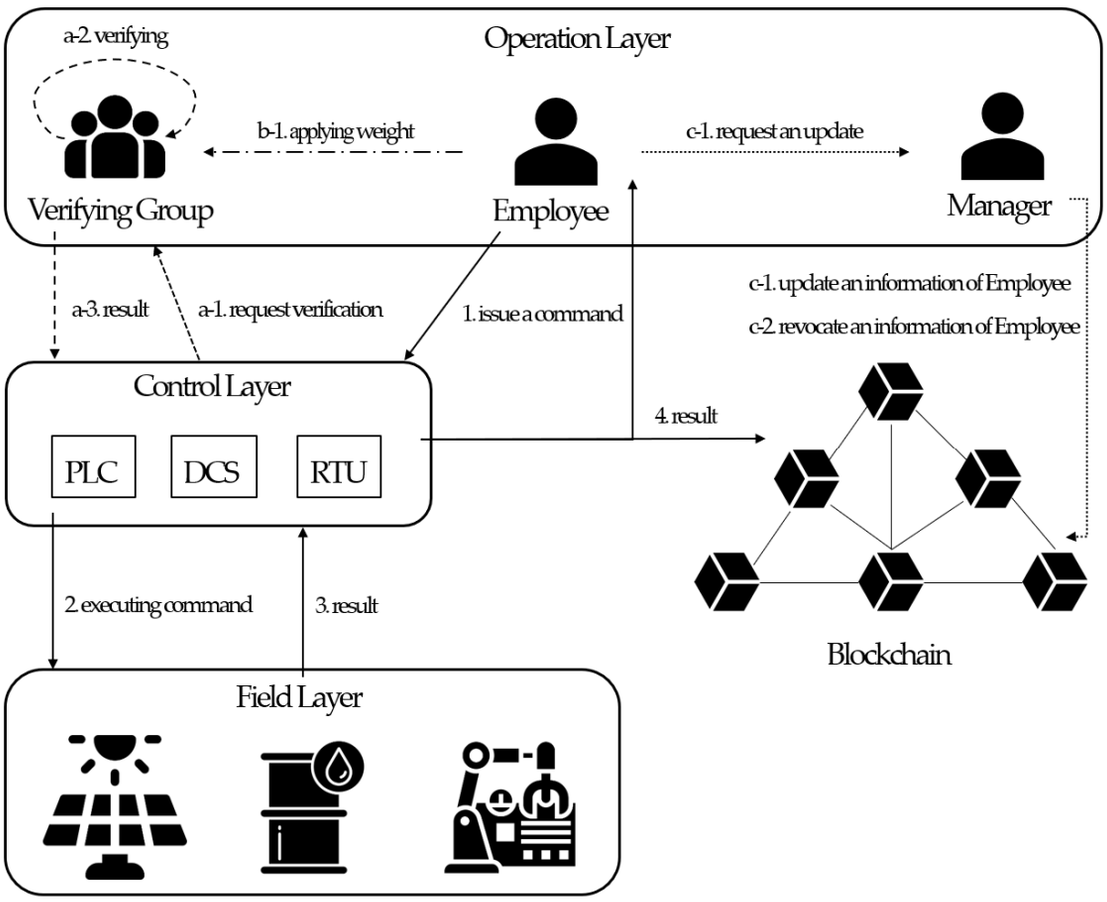

Fig. 5. Structure of Proposed System

이도를 이용해 생성할 수 있지만, 블록이 확정된 후에 해당 값을 이용할 수 있어 발생시간이 불규칙하거나 오래 걸릴 수 있다. 본 논문에서 제안하는 랜덤리스트는 블록의 확정과 관계없이 리스트를 난수를 이용해 섞는 연산으로 랜덤리스트를 생성하기 때문에 블록의 타임스탬프나 난이도 값 사용을 위해 지연되는 시간을 줄일 수 있고, 다른 방식들과 비교해 연산이 간단하다.

Fig. 5는 제안시스템의 구조를 나타낸다. 시스템은 일반적인 산업제어시스템과 같이 운영계층, 제어계층, 현장계층으로 구성되어있다. 각 계층에서 발생하는 연산의 결과는 블록체인에 기록되어 데이터의 무결성을 만족할 수 있도록 한다. 운영계층은 제어계층에게 현장계층을 제어하기 위한 명령을 내린다. 제어계층은 운영계층으로부터 전달받은 명령을 검증하기 위한 프로세스를 가지며, 검증 결과에 따라 현장계층을 제어한다. 검증 프로세스는 운영계층이 내린 명령을 현장계층에서 실행해도 되는지 명령의 유효성을 확인하는 과정이다. 검증의 주체는 검증그룹이며, 명령을 내린 직원을 제외한 나머지 직원으로 구성된다. 검증그룹을 구성하기 위해 계약 배포 시 생성한 랜덤리스트를 사용한다. 현장계층은 운영계층으로부터 제어계층이 내린 명령을 수행하거나 필요한 데이터를 수집해 제어계층으로 전달한다.

3.2 인적 자원 관리 스마트 계약

인적 자원 관리 스마트 계약은 모든 직원의 정보, 권한 등을 관리하는 계약이다. 이때 권한 관리는 관리자로 선정된 직원만 할 수 있고, 주로 인사 담당자와 같이 직급이 높은 직원이 맡도록 한다. 인적 자원 관리 스마트 계약의 핵심은 관리자와 블록체인을 이용해 직원 정보에 대한 갱신 요청이 있을 때 실시간으로 정보를 수정하는 데 불편함이 없게 하는 것이며, 직원의 정보가 변경되거나, 이직, 퇴직 등의 경우로 직원의 계정이 불필요할 경우 즉각적으로 조치할 수 있게 한다.

갱신/폐기 요청 트랜잭션은 Fig. 6의 1번에 해당한다. 직원은 매니저에게 자신의 공개키와 직원 일련번호를 이용해 트랜잭션을 발생시켜 권한 조정을 요청한다. 이때 직원의 키는 본인이 생성하고 개인 키는 직원 자신만 알아야 한다[7].

갱신 트랜잭션은 Fig. 6의 a에 해당한다. 갱신 요청 트랜잭션을 확인한 관리자가 트랜잭션을 발생시킨 직원의 공개키, 직원 일련번호 등을 확인하고 이를 파라미터로 사용해 해당 직원의 정보를 갱신한다. 갱신한 내용은 블록체인에 기록되며 갱신 내용은 직원에게 새로운 권한을 부여하거나 기존에 갖고 있던 권한을 박탈하는 것이다[7].

폐기 트랜잭션은 Fig. 6의 b에 해당한다. 직원이 회사에서 퇴사하거나, 다른 직장으로 이직을 했을 경우 등 기존 직원의

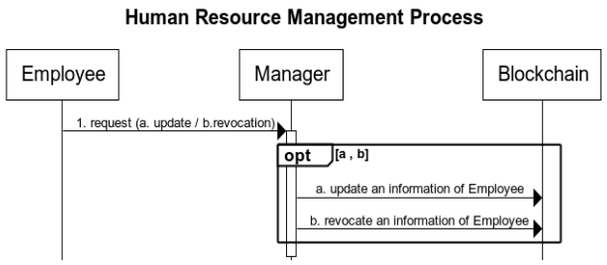

Fig. 6. Sequence Diagram of Human Resource Management Process

계정이 존재할 이유가 없을 때 발생시키는 트랜잭션이다. 폐기 트랜잭션도 직원 일련번호를 파라미터로 전달한다. 만약 이직 혹은 퇴직한 직원이 회사로 돌아오는 경우 새로운 공개키와 비밀키를 발급하고 새로운 직원 일련번호를 할당받은 후, 갱신 트랜잭션을 통해 권한을 부여받아야 한다.

3.3 운용 로그 스마트 계약

산업제어시스템에서 중요한 취약점 중 하나는 책임 추적성을 만족하지 못하는 것이다. 책임 추적성을 만족하지 못하는 산업제어시스템의 경우 오류나 사고 발생 시 원인 파악이 매우 어려울 뿐만 아니라, 기록에 대한 무결성을 보장하기 어렵고 산업제어시스템 전체에 매우 치명적인 영향을 끼칠 수 있다. 운용 로그 스마트 계약을 통해 명령 수행 과정을 블록체인에 기록함으로써 책임 추적성을 보장하며 명령 수행 기록의 무결성 또한 확보할 수 있다. 제안시스템에서는 운용 로그 스마트 계약을 배포하는 과정에서 계약의 생성자 함수를 이용해 검증그룹을 선정하기 위한 랜덤리스트를 생성한다.

명령 트랜잭션은 Fig. 7의 1번에 해당하며 운영계층에서 제어계층으로 명령을 내리고자 할 때 발생시킨다. 운영계층에 속한 직원이 명령을 내릴 때 제어 명령, 직원의 비밀키를 활용한 서명, 직원 식별번호 등을 입력으로 받아 명령을 내린다.

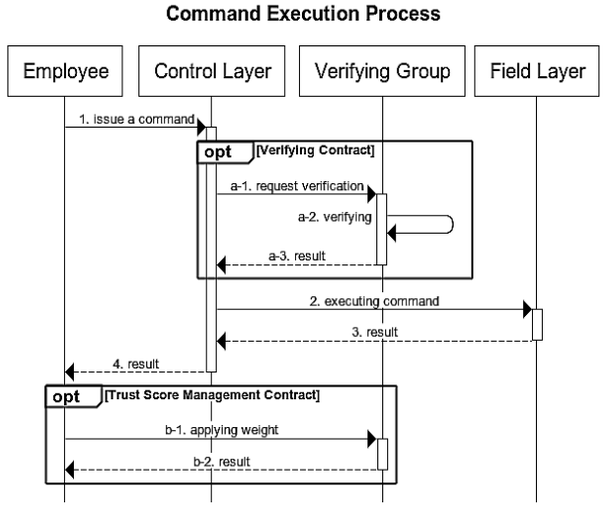

Fig. 7. Sequence Diagram of Command Execution Process

Condition: Verify when Command Score >= 5

Command Number	Content	Command Score	Need Verification
1	System Reboot	10	O
2	System Update	9	O
3	System Check	2	X
4	Update Firewall Policy	7	O
5	Sensor Check	1	X

Fig. 8. Example of Commands

명령 트랜잭션이 발생하면, 제어계층은 발생한 명령 트랜잭션의 제어 명령이 검증이 필요한 명령인지 확인 후, 검증이 필요한 명령일 경우 해당 명령에 대한 검증 스마트 계약을 발행해 운영계층에 명령 검증을 요청한다. 검증이 필요하지 않은 명령의 경우 검증 과정을 생략함으로써 불필요한 연산을 줄이고, 시스템에 가용성을 부여한다. 검증이 필요한 명령과 검증이 필요하지 않은 명령은 산업제어시스템을 관리하는 주체가 정한 기준에 따라 결정된다. 일반적으로 검증이 필요한 명령은 실행되었을 때 전체 시스템에 영향을 줄 수 있는 명령이며, 검증이 필요하지 않은 명령은 실행되었을 때 시스템에 영향을 주지 않는 명령이다. 시스템에 큰 영향을 줄 수 있는 명령일수록 명령을 실행하기 위한 명령 점수가 크고, 반대로 시스템에 영향을 줄 수 없는 명령일수록 명령 점수가 작다.

Fig. 8은 검증을 진행하는 명령과 검증을 진행하지 않는 명령을 그림으로 나타낸 예시다. 명령 점수가 5점과 같거나 큰 경우에만 검증을 진행하라는 조건이 있고, 표에 있는 명령 1, 2, 4번에 해당한다. 해당 명령들은 시스템 재부팅, 시스템 업데이트, 방화벽 정책 업데이트 등 시스템에 큰 영향을 끼칠 수 있는 명령들이며 각 명령 사이에도 영향도를 기준으로 한 점수 차이가 존재한다. 반면, 3번과 5번 명령과 같이 시스템 상태를 확인하거나 센서의 상태를 확인하는 명령은 시스템에 큰 영향을 주지 않기 때문에 명령 점수도 낮게 책정되어 있으며 검증 과정을 거치지 않는다.

3.4 검증 스마트 계약

산업제어시스템에서의 명령 실행 절차는 매우 까다로워야 하며 꼭 필요한 경우에만 실행되어야 한다. 검증 스마트 계약은 Fig. 7의 a-부분이며, 명령에 따라 선택적(Fig. 7. Optional a)으로 발생한다. 만약 운영계층의 직원이 중요성이 낮아 검증이 불필요한 명령을 내렸을 경우 해당 명령은 검증 스마트 계약을 발생시키지 않고 곧바로 현장계층에 실행되어 불필요한 자원 낭비를 줄일 수 있도록 한다. 중요성이 높아 검증이 필요한 명령의 경우, 제어계층은 명령 점수를 파라미터로 이용해 운영계층에 속해있는 직원 중 명령을 검증해줄 검증그룹을 선정해 명령 검증을 요청한다.

검증요청 트랜잭션은 FIg. 7의 a-1에 해당하며, 명령 트랜잭션을 확인한 제어계층이 명령 점수를 이용해 발생시킨다. 명령을 내린 직원을 제외한 직원 중, 3.3에서 생성한 랜덤리

스트를 이용해 검증그룹을 구성하여 시행된 명령에 대해 검증을 요청하는 트랜잭션이다. 검증그룹은 랜덤리스트에 담겨있는 순서로 구성되며, 가용성 확보를 위해 명령 실행에 필요한 점수의 2배수를 만족하는 직원들로 구성되어있다. 예를 들어 직원이 내린 명령 점수가 5점일 경우, 점수의 2배수인 10점을 만족하는 직원을 검증그룹으로 구성해 명령에 대해 검증을 요청한다. 이때 검증그룹에 속해있는 직원들은 최대한 다양한 직급의 인원으로 구성할 수 있도록 하여 높은 직급에 있는 공격자가 시스템을 쉽게 장악할 수 없도록 한다.

검증 트랜잭션은 Fig. 7의 a-2에 해당하며 랜덤리스트로부터 선정되어 검증요청을 받은 검증그룹이 명령을 확인한 후 검증하는 트랜잭션이다. 검증요청을 받은 검증그룹은 명령이 올바르게 내려진 명령인지, 시스템에 악영향을 끼치지 않을지 등의 판단 후 개인 키를 이용한 서명, 직원 식별번호를 이용해 명령에 대한 검증, 일종의 OK 사인을 내린다.

예를 들어 명령 실행에 필요한 점수가 5점이고, 제어계층에서 점수의 2배수만큼 10점을 만족하는 검증그룹에 검증을 요청한 상황을 가정해본다. 이 경우 검증그룹에 속해있는 각 개인이 판단하여 검증그룹 중 일부 직원은 명령 실행이 올바르다고 판단하여 검증을 진행할 수 있고, 반대로 다른 일부는 명령 실행이 올바르지 않다고 판단하여 검증을 진행하지 않을 수 있다, 이 같은 경우 명령을 검증한 직원의 점수 합계가 명령 실행에 필요한 점수인 5점과 같거나 그 이상이라면 해당 명령은 검증 과정을 마치고 정상적으로 실행된다.

비슷한 맥락으로 검증그룹의 최소 검증 인원을 설정해 명령 실행 점수를 만족하더라도 검증그룹에 속한 최소 n명이 검증해야 실행될 수 있게끔 하는 방법도 선택적으로 사용할 수 있다. 그룹 내에 점수가 높은 몇 명이 나쁜 마음을 먹고 시스템을 공격하기 위한 목적으로 고의로 잘못된 명령을 검증해 명령이 수행됨을 방지하기 위함이다.

검증 과정에서 시스템의 가용성을 추가하기 위해 검증을 진행한 직원들의 직원 신뢰 점수를 합산해, 명령 점수의 2배수가 넘으면 해당 명령 검증 과정에 양의 가중치를 부여할 수 있다. 직원 신뢰 점수는 직급마다 차이가 없고, 검증그룹에 속해 명령을 검증했을 경우, 검증 결과가 긍정적이면 양의 가중치를 부여하고 부정적이면 음의 가중치를 부여한다. 가중치의 정도는 산업제어시스템을 관리하는 주체에서 정한 기준을 따른다.

실행 결과 트랜잭션은 Fig. 7의 3번, 4번에 해당하며 현장계층에서 명령을 실행한 뒤, 명령의 실행 결과를 제어계층으로 전달하고 이어서 제어계층이 운영계층으로 전달한다. 실행 결과 트랜잭션을 통해 해당 명령 실행의 영향, 성공 여부 등을 확인할 수 있으며, 책임 추적성을 만족할 수 있도록 하는 역할도 한다. 실행 결과 트랜잭션의 결과는 신뢰 점수 관리 스마트 계약의 핵심적인 지표로써 사용된다.

3.5 신뢰 점수 관리 스마트 계약

신뢰 점수 관리 스마트 계약은 Fig. 7의 b-부분에 해당하

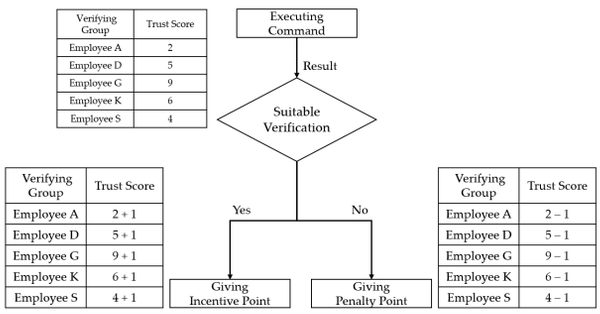

Fig. 9. Trust Score Management Mechanism

며, 검증 프로세스와 같이 선택적(Fig. 7. Optional b)으로 발생한다. 신뢰 점수란 직원을 관리하고 직원이 일을 올바르게 하고 있는지 판단할 수 있는 지표로써 명령을 검증할 때도 사용되는 직원의 점수를 말한다. 신뢰 점수는 직원의 직급과 관계없이 기본값이 같으며 일정 주기를 기준으로 초기화된다. 검증그룹으로부터 검증된 명령이 실행된 후 현장계층이 넘겨준 실행 결과에 따라 검증그룹에 속한 직원들의 신뢰 점수에 피드백을 부여한다.

가중치 적용 트랜잭션은 Fig. 7의 b-1에 해당한다. 실행 결과 트랜잭션에서 반환된 결과를 참조하여 검증그룹에 속해있는 직원들의 신뢰 점수에 가중치를 적용한다. 검증그룹으로부터 검증되어 실행된 명령이 유효하고 정상적인 행위였다면 양의 가중치를 적용하고, 불필요하거나 정상적이지 않은 명령이었을 경우 음의 가중치를 적용한다. 피드백을 통한 신뢰 점수의 가중치 적용 과정을 통해 직원을 평가할 수 있는 지표로서 가치를 가질 수 있다.

Fig. 9는 신뢰 점수 관리 메커니즘을 나타낸다. 그림은 명령 실행 후 결과가 도출되었을 때의 상황만을 담고 있으며 명령에 대한 검증그룹으로 5명의 직원이 선택되었다. 명령 검증 행위가 적절할 경우 검증그룹에 속해있는 직원에게 모두 인센티브 점수 1점을 부여한다. 반대로, 적절하지 않을 경우 페널티 점수 1점을 부여한다. 인센티브 점수나 페널티 점수는 산업제어시스템을 관리하는 주체가 자체적으로 설정할 수 있으며, 신뢰 점수가 n점 이하인 직원은 검증그룹에 포함될 수 없도록 설정해 명령 검증 과정에 신뢰성과 정확성을 부여할 수 있다.

3.6 데이터 전달 스마트 계약

산업제어시스템에서 명령을 내리거나, 검증요청을 하거나, 직원의 권한을 갱신/폐지할 때 등 블록체인과 스마트 계약을 이용한 모든 활동은 데이터를 주고받으며 상호작용한다. 하지만 별다른 조치 없이 데이터를 주고받는다면, 네트워크 통신을 조작하여 데이터를 엿듣거나 데이터를 조작하는 중간자 공격 등으로부터 데이터의 무결성을 보장할 수 없다.

1) 데이터 해시 트랜잭션

제안시스템에서는 네트워크 모델의 체크섬이나 블록체인

의 이전 블록 해시값 확인 과정 등과 비슷한 개념으로, 특정 주체에게 전달할 데이터를 해시 함수의 입력값으로 넣어 결과로 도출된 해시값을 함께 전달한다. 이러한 과정을 통해 받는 쪽에서는 해시 함수를 이용해 데이터를 검사하여 정상적인 데이터를 수신했는지 확인할 수 있다. 해시 함수는 SHA, Keccak 등을 이용할 수 있으며, 산업제어시스템의 유형 중, 실시간성에 여유가 있는 시스템은 RSA 알고리즘 등을 사용하여 더욱 강력한 데이터 전달 보안을 보장할 수 있다.

4. 성능 평가

4.1 실험 환경 구성

제안시스템과 기존시스템[7]의 비교를 위해 양측 시스템을 이더리움의 튜링 완전 언어인 솔리디티를 통해 구현했으며, 이더리움 엔진인 Geth[17]를 통해 사설 네트워크를 구축했다. 솔리디티로 작성한 소스코드를 컴파일러를 통해 ABI (Application Binary Interface), Bin(Binary) 데이터로 변환하여 스마트 계약을 배포하기 위한 자바스크립트 파일을 작성했으며, web3[18] 자바스크립트 라이브러리를 이용해 Geth로 구축한 사설 네트워크에 배포했다. 스마트 계약 배포와 트랜잭션 연산을 위한 채굴 시, 쓰레드 4개를 사용하여 채굴했으며 사설 네트워크의 블록 채굴 난이도는 '0x10'으로 낮게 설정하여 배포 시 채굴 작업에 걸리는 지연시간을 최소화했다.

4.2 실험결과 및 분석

가용성은 서버나 프로그램, 네트워크 등의 시스템이 정상 범위 내에서 사용 가능한 정도를 나타내는 지표로, 산업제어시스템의 주요 속성이며 산업제어시스템의 구성이나 종류에 따라 다른 속성보다 중요할 때가 있다. 이더리움의 가스는 스마트 계약을 배포하거나 트랜잭션을 발행할 때 필요한 컴퓨팅 에너지로, 가스가 모두 소모될 경우 새로운 트랜잭션을 발행할 수 없다. 이는 곧 가스 사용량이 많을수록, 사용 속도가 빠를수록 시스템의 가용성이 떨어진다고 할 수 있다.

본 논문에서는 기존시스템과 제안시스템의 스마트 계약 배포 과정과 검증 과정에서 사용되는 수수료인 가스 사용량을 비교했다. 이더리움 네트워크의 가스는 측정된 가스 사용량으로부터 가용성을 비교하기 위해 Fig. 11과 Fig. 12에 주목한다. 가스 사용량은 코드가 변하지 않는 이상 정량적으로 측정할 수 있는 값이며, 추후 코드가 변경되거나 내용이 추가될 경우 늘어나거나 줄어들 수 있다. 또한, Fig. 11과 Fig. 12에서 검증이 필요하지 않은 명령 수행은 단순히 운영계층이 내린 명령 전달에 가까워 가스 사용량 비교의 의미가 없다고 판단해 검증이 필요한 명령만 실험을 진행했다.

1) 명령 실행시간 비교

Fig. 10은 4.1의 실험 환경에서 양측 시스템이 명령을 내리기 위한 스마트 계약 배포에 걸리는 시간을 측정했다.

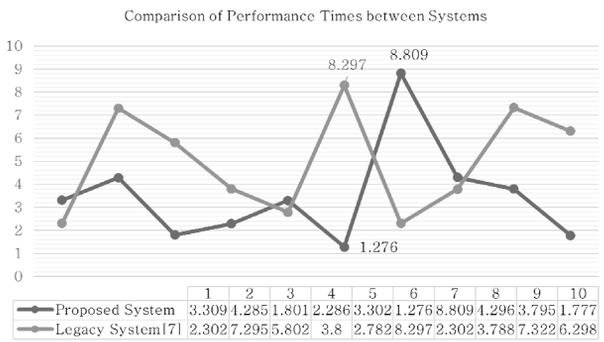

Fig. 10. Required Time for Deploying Command Contract

파란색 꺾은선 그래프가 제안시스템을 뜻하며, 주황색 꺾은선 그래프가 기존시스템을 뜻한다. 그래프 아래 표는 실험 10회 측정 결과이며, 결과의 평균으로 제안시스템은 3.4936초를 기록했으며 기존시스템은 4.9988초를 기록했다. 기존시스템의 6회차나, 제안시스템의 6회, 7회와 같은 결과는 채굴 과정의 지연으로 인한 이상치로써 판단할 수 있다. 제안시스템에서는 스마트 계약 배포 시 랜덤리스트를 생성하기 위한 연산을 진행하지만, 기존시스템과 비교해 배포 시 걸리는 시간에 큰 차이가 없음을 확인했다.

2) 명령 배포 과정의 가스 사용량 비교

Fig. 11은 두 시스템의 스마트 계약 배포 시 가스 사용량을 측정했다. 제안시스템은 2,282,947의 가스를 사용했고, 기존시스템은 1,621,647의 가스를 사용했다. 제안시스템은 기존시스템과 비교해 랜덤 구현 방법을 구체화하여, 스마트 계약을 배포할 때 생성자를 이용해 직원 수 길이만큼의 랜덤리스트를 생성한다. 생성자를 통해 랜덤리스트를 생성하는 과정의 연산으로부터 생긴 오버헤드로 시스템 간 가스 사용량의 차이가 도출되었다. 즉, 검증 요청과정에 필요한 랜덤리스트를 생성하는 과정에서 가스 사용량의 차이가 발생했다. 본 논문에서 제안하는 랜덤리스트는 명령 검증 요청과정에서 랜덤을 생성하는 것보다 스마트 계약을 배포할 때 가스는 많이 사용하지만, 검증 과정을 최적화할 수 있다.

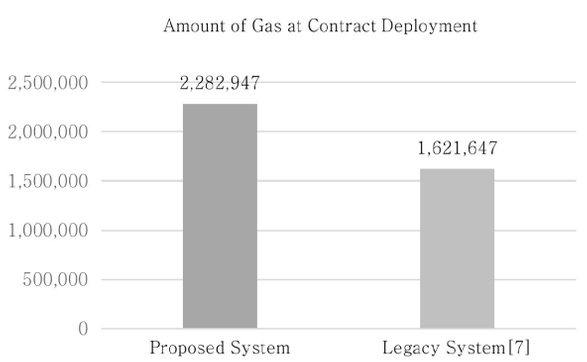

Fig. 11. Amount of Gas Deploying Contract

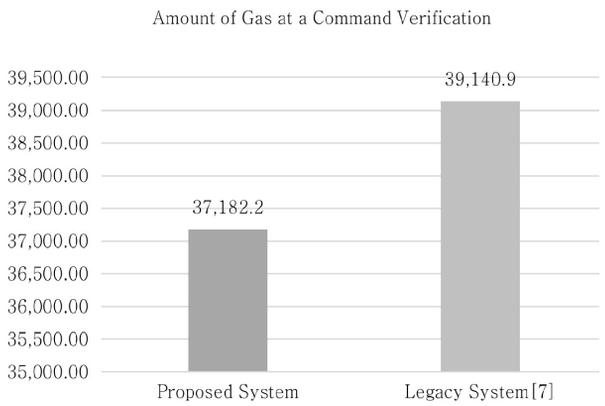

Fig. 12. Amount of Gas Verifying Command

3) 검증 과정의 가스 사용량 비교

Fig. 12는 두 시스템의 검증 1회에 사용되는 가스 사용량을 측정했다. 검증 과정에 랜덤리스트를 사용하는 제안시스템은 기존시스템과 같이 검증 과정에서 랜덤을 매번 생성하지 않아 적은 오버헤드가 측정되었다. 각 시스템의 차이는 검증을 위한 랜덤 구현 방법으로부터 도출된다. 기존시스템의 경우 검증을 진행할 때마다 새로운 랜덤을 이용하기 때문에 랜덤리스트를 사용하는 제안시스템에 비해 더 많은 연산이 필요하고, 그에 따라 더 많은 가스를 사용하게 된다. 그와 반대로 제안시스템은 랜덤리스트로부터 값을 불러오는 연산만 필요하므로 오버헤드가 적지만, 신뢰 점수 스마트 계약과 관련된 연산을 검증 중에 수행하기 때문에 약간의 오버헤드가 추가된 가스 사용량이 측정되었다. 신뢰 점수 스마트 계약을 이용해 발생한 오버헤드는 검증 과정에 가용성을 부여할 수 있어, 불필요한 오버헤드는 아니다.

스마트 계약 배포 시 가스 사용량, 명령 검증 시 가스 사용량을 합쳐보면 기존시스템과 비교해 제안시스템의 가스 사용량이 많은 것으로 해석될 가능성이 있다. 기존시스템은 스마트 계약을 배포할 때 랜덤리스트를 생성하지 않아 작은 가스를 사용하지만, 검증 과정에서 명령을 검증할 때마다 새로운 난수를 생성해야 하는 문제로 인해 추가적인 오버헤드가 검증 과정 중 계속 발생하며, 랜덤을 구현하는 방식에 따라 추가 오버헤드가 발생할 수 있는 문제가 있다. 하지만, 제안시스템은 스마트 계약을 배포할 때, 많은 가스를 사용해 검증 과정에 사용될 랜덤리스트를 미리 생성하여 검증 과정에 필요한 랜덤을 만족하도록 했다. 이를 통해 검증 과정 중 난수를 생성하지 않아 불필요한 가스의 낭비를 줄여 가용성을 만족할 수 있도록 하며, 추가적인 가용성 확보를 위한 신뢰 점수 메커니즘 연산을 수행할 수 있도록 했다.

5. 결 론

폐쇄적으로 운영되고 있던 산업제어시스템은 인터넷의 발전과 ICT 기술의 도입으로 꾸준히 발전하고 종류 또한 다양해지고 있다. 하지만, 그에 따라 외부에 노출되는 일이 많아졌고 인터넷을 통해 손쉽게 접근할 수 있어 공격자들의 보안 위협 또한 증가했다. 발전 시설, 다양한 공장 산업 등을 구성하고 있는 산업제어시스템에 대한 보안 위협은 목적에 맞는 일정한 상태를 유지해야 하는 산업제어시스템에 치명적이며, 이는 방대한 경제적 손실과 인명 피해를 일으킬 수 있어 특별한 주의를 기울여야 한다. 또한, 산업제어시스템의 가용성 침해는 보안 위협과 비슷하게 심각한 피해를 불러올 수 있기에, 어떤 경우에서도 침해되지 말아야 하며, 시스템의 가용성을 유지하기 위해 산업제어시스템은 적절한 접근제어를 통해 외부의 공격이나 방해로부터 산업제어시스템을 방어해야 한다.

본 논문에서는 난수와 모듈러 연산을 이용해 생성한 랜덤리스트를 이용하여 블록체인 기반의 산업제어시스템 접근제어 방식을 새롭게 제안했다. 또한, 검증그룹의 명령 검증 행위에 책임감과 신뢰성을 부여할 수 있도록 하는 직원 신뢰 점수 메커니즘을 새롭게 도입하여 인센티브나 벌점을 부여하고 직원 평가지표로써 사용하고 검증 과정에 추가요인으로 작용하여 유연성과 가용성을 부여할 수 있도록 했다.

향후 연구에서는 조금 더 공정하고 오버헤드가 적은 랜덤리스트를 만드는 방법을 고안하고 데이터 전달 과정에 기밀성과 무결성을 함께 보장하는 방안을 알아보며, 사람이 수작업으로 검증하지 않아도 자동으로 명령 적합성을 판단하여 자체적인 검증 과정을 통해 명령을 실행할 수 있도록 하는 인공지능 시스템의 도입하고자 한다. 또한, 블록체인을 이용한 산업 IoT 환경에서의 접근제어 기법들과 제안시스템의 성능 비교를 통해 스마트 계약의 설계나 구현 부분을 검토하고 개선하여 시스템의 성능을 최적화할 계획이다.

References

[1] J. H. Nah and J. C. Nah, “Standardization trend of industrial control system security,” *Review of Korea Institute of Information Security & Cryptology (KIISC)*, Vol.26, No.4, pp.28-35, 2016.

[2] J. H. Oh, Y. I. You, and K. H. Lee, “Infrastructure incident and control system standard trend,” *Review of Korea Institute of Information Security & Cryptology (KIISC)*, Vol.27, No.2, pp.5-11, 2017.

[3] S. Keith and P. Victoria, “Guide to industrial control systems (ICS) security,” in *NIST Special Publication*, 800-82, 2015.

[4] K. H. Kim, “Industrial control system security,” in Institute for Information & Communication Technology Planning & Evaluation(IITP) Weekly ICT Trends, pp.2-14, 2021.

[5] M. K. Kang, “CyberSecurity status by country for industrial control system,” in Institute for Information & Communication Technology Planning & Evaluation(IITP) Weekly ICT Trends, pp.16-24, 2019.

[6] M. Mao and H. Xiao, "Blockchain-based technology for industrial control system cypersecurity," in International Conference on Network, Communication, Computer Engineering, pp.2-5, 2018.
[7] M. J. Cho and C. H. Lee, "Access control mechanism in industrial control system based on smart contract," *Review of Korea Institute of Information Security & Cryptology (KIISC)*, Vol.29, No.3, pp.579-588, 2019.
[8] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, 2008.
[9] V. Buterin, "Ethereum white paper: A next-generation smart contract and decentralized application platform," *White Paper*, Vol.3, No.37, pp.1-36, 2014.
[10] M. S. Ferdous, M. J. M. Chowdhury, M. A. Hoque, and A. Colman, "Blockchain consensus algorithms: A survey," *arXiv preprint arXiv:2001.07091*, 2020.
[11] S. Nick, "Smart contracts: Building blocks for digital markets," *EXTROPY: The Journal of Transhumanist Thought*, Vol.18, No.2, pp.28, 1996.
[12] Solidity [Internet], https://docs.soliditylang.org/en/v0.8.2/, 2021.
[13] J. W. Kim, "Legal Issues of the 'Smart Contract'," *Korea Lawyers Association Journal (KLAJ)*, Vol.67, No.1, pp.150-200, 2018.
[14] H. S. Kim, "Blockchain-based smart contract and legal issues," *Dankook Law Review*, Vol.44, pp.171-192, 2020.
[15] T. Sultana, A. Almogren, M. Akbar, M. Zuair, I. Ullah, and N. Javaid, "Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT Devices," *Applied Sciences*, Vol.10, No.2, pp.488-509, 2020.
[16] H. Albreiki, L. Alqassem, K. Salah, et. al. "Decentralized access control for IoT data using blockchain and trusted oracles," *IEEE International Conference on Industrial Internet (ICII)*, pp.248-257, 2019.
[17] Geth [Internet], https://geth.ethereum.org/docs/.
[18] web3 [Internet], https://web3js.readthedocs.io/en/v1.3.4/.

강 명 조

https://orcid.org/0000-0002-0691-2970
e-mail : rkdaudwh13@hknu.ac.kr
2021년 한경대학교 컴퓨터응용수학부(학사)
2022년 ~ 현 재 한경대학교 컴퓨터응용수학부 석사과정
관심분야 : 네트워크 보안, 블록체인, 머신 러닝

김 미 희

https://orcid.org/0000-0002-4896-7400
e-mail : mhkim@hknu.ac.kr
1997년 이화여자대학교 전자계산학과 (학사)
1999년 이화여자대학교 컴퓨터학과(석사)
1999년 ~ 2003년 한국전자통신연구원 연구원
2007년 이화여자대학교 컴퓨터학과(박사)
2007년 ~ 2009년 이화여자대학교 컴퓨터학과 전임강사
2009년 ~ 2010년 노스캐롤라이나주립대학교 연구원
2011년 ~ 현 재 한경대학교 컴퓨터응용수학부 컴퓨터시스템연구소 교수
관심분야 : 네트워크 성능 분석 및 보안, 무선네트워크 보안, 침입대응, 크라우드센싱, 블록체인