ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 19-04-33
228

j.inst.Korean.electr.electron.eng.Vol.23,No.4,1343~1352,December 2019
http://dx.doi.org/10.7471/ikeee.2019.23.4.1343

블록체인 기반 안전한 사물인터넷 장치 관리 시스템 구현

Implementing Blockchain Based Secure IoT Device Management System

김 미 희*★, 김 영 민*

Mihui Kim*★, Youngmin Kim*

Abstract

To manage the Internet of Things(IoT) Network, which consists of a large number of various devices, a secure and automatic method of strengthening the IoT network is being proposed. Blockchain has a 'smart contract' element of autonomous execution method, which is emerging as a way to not only exchange data quickly without mediators but also securely and automatically manage processes between IoT devices. In this paper, we implement a prototype of the entire IoT device management system based on the EOSIO with DPoS(Distributed Proof of Stake)-based blockchain structure, proposed as a prior study, including the user application DApp(Decentralized Application) and the actual IoT devices (Raspberry Pi-based device, and smart lamp) that interact with the blockchain platform. We analyze the benefits of the system and measure the time overhead to show the feasibility of the system.

요 약

많은 수의 다양한 기기로 구성된 사물인터넷(Internet of Things, IoT)의 기기 관리를 위해서 안전하고 자동으로 강화하는 방법을 강구하고 있다. 블록체인은 자율 실행 방식의 '스마트 컨트랙트' 요소로 중재자 없이 빠르게 데이터를 교환하고 IoT 디바이스 간 프로세스를 안전하고 자동으로 관리할 수 있는 방안으로 대두되고 있다. 본 논문에서는 선행연구로서 제안한 DPoS(Distributed Proof of Stake) 체계인 EOSIO 기반 블록체인 구조를 바탕으로 블록체인 플랫폼과 상호작용 하는 사용자 응용 DApp(Decentralized Application)과 실제 IoT 기기(라즈베리파이 기반 기기, 스마트 램프)를 포함하여 전체 시스템의 프로토타입을 구현한다. 시스템의 이점을 분석하고, 시간적 측면의 오버헤드를 측정하여 본 시스템의 실현 가능성 보이고자 한다.

Key words : IoT device management, Blockchain, System Prototype Implementation, DApp, Smart Contract

* Dept. of Computer Science & Eng., Computer System Institute, Hankyong National University
★ Corresponding author
E-mail : mhkim@hknu.ac.kr, Tel : +82-31-670-5167
※ This research was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No.2018R1A2B6009620).
Manuscript received Dec. 12, 2019; revised Dec. 20, 2019; accepted Dec. 26, 2019.

Ⅰ. 서론

최근 자율자동차, 의료기기, 공장자동화, 스마트 시티에서의 건물 및 도시 인프라 관리를 위해 센서 기반 사물인터넷(Internet of Things, IoT) 기기들의 수가 급증하고 있고, 2022년에는 2.5배 증가하여 전 세계 네트워크 기기의 절반을 차지할 것으로 예상하고 있다[1]. 정부와 기업에서는 이러한 많은 수의 IoT 기기에서 만들어지는 트랜잭션 프로세스를 처리하기 위해 네트워크를 안전하고 자동으로 강화하는 방법을 강구하고 있다[2]. 이는 IoT 기기들

이 보안 표준이 견고하지 않고, 그마저 지키지 않는 제품과 서비스 회사들이 많기 때문에 이러한 취약점을 노린 공격이 확대될 수 있다고 경고하고 있기 때문이다. 실제 2019년 3월 미국 애틀란타 시가 랜섬웨어 공격을 받아 1,700만 달러에 달하는 피해를 입기도 했고, 스마트시티의 취약한 IoT 기기들을 대상으로 하는 공격이 더욱 증가할 것으로 예측하고 있다[3].

이러한 많은 수의 보안상 취약한 그래서 안전성과 자동화가 중요한 IoT 기기 관리를 위해 블록체인 기술이 하나의 방안으로서 대두되고 있다. 분산원장 기술(Distributed Ledger Technology)인 블록체인은 자율 실행 방식의 '스마트 컨트랙트'로 알려진 비즈니스 자동화 소프트웨어 요소를 포함하고 있어서 중재자 없이 빠르게 데이터를 교환하고 IoT 디바이스 간 프로세스를 안전하게 강화하는 표준화된 방법과 방식을 제공한다[4]. 예를 들어, IoT 기기의 인증 및 접근제어 프로세스를 스마트 컨트랙트로 구현함으로써 안전한 체계를 서버 없이 구현할 수 있다. 이러한 스마트 컨트랙트의 실행 결과는 블록에 기록되어 연결되므로 무결성을 제공하여 그 안전성을 높일 수 있다.

IoT 관리 네트워크를 위해 블록체인 기술을 접목하려고 하는 시도가 학계, 산업계에서 진행 중이다. [5]의 논문에서는 각기 다른 IoT 장치의 사용자 인증 방법의 다양화로 인한 연동 문제를 해결하기 위해 블록체인을 활용하여 여러 IoT 네트워크에서의 사용자 인증을 통일하는 기법을 제안했다. [6]에서는 홈 IoT 관리를 위해서 PoW(Proof of Work, 작업 증명) 합의 알고리즘 기반 스마트 컨트랙트가 동작하는 네트워크를 제안하였다. 연구 [7]에서는 차량 자산관리, 스마트 교통관리, 물류관리를 위해 IoT와 블록체인 기술을 결합한 기본적인 구조를 제안하였다. 연구 [8]에서는 IoT 기반 스마트 시티를 구성할 때, 스마트 차의 참여를 가지고 무선센서네트워크를 구성하는데 롤링 블록체인 구조를 제안하였다. 이 연구들은 아직 블록체인 PoW 알고리즘을 그대로 적용하는데서 오는 확장성과 처리 속도의 제약이 있거나, 기본적인 구조만 제시하고 있어 구현으로서 실현가능성을 보이고 있지는 못하다.

안전한 IoT 기기 관리를 위해서, 선행 연구[9]에서는 PoW 합의 알고리즘이 아닌 DPoS(Distributed Proof of Stake, 분산지분증명) 합의 알고리즘을 활용함으로써 실행 속도와 확장성을 개선한 새로운 블록체인 IoT 관리 구조를 제안하였다. 제안된 구조에서는 PoW와 비교해서 얻을 수 있는 확장성과 트랜잭션의 처리 속도의 이점을 실험을 통해 보였다. 본 논문에서는 선행 연구인 이러한 구조를 바탕으로 블록체인 플랫폼과 상호작용 하는 자바 기반 모바일 애플리케이션 DApp(Decentralized Application)과 실제 IoT 기기(라즈베리파이 기반 기기, 스마트 램프)를 포함하여 전체 시스템의 프로토타입을 구현하였다. 사용자 입장에서 실제 IoT 기기를 DApp으로 동작시키고 블록체인을 통해 관리하는 이점을 분석하고, 블록체인 기반 관리로 인한 시간적인 오버헤드를 측정해서 본 시스템의 실현 가능성 보이고자 한다. 본 논문의 기여부분은 다음과 같다.

- 블록체인 플랫폼, DApp, IoT 기기 포함 전체 시스템 프로토타입 구현
- 구현 시스템 구동 시험 및 성능 측정

2장에서는 안전한 관리를 요구하는 IoT 기반 시스템과, 기본적인 블록체인 개념에 대해 설명하고, 3장에서는 기반 되는 블록체인 구조를 설명한다. 4장에서는 제안 시스템 프토토타입 구현 내용을 설명하고, 5장에서 실험을 통해 제안 시스템의 성능을 분석한다. 마지막으로 6장에서 결론을 맺는다.

II. 기반 연구

1. 안전한 관리 요구하는 IoT 기반 시스템

스마트 홈은 조명 장치, 난방 장치 등을 인터넷을 통해 원격으로 제어 또는 모니터링 하거나 작동을 자동화 할 수 있도록 돕는 시스템이다. 이러한 스마트 홈 시스템이 다루는 데이터는 가정 내에서 수집하거나 생성되는 것으로, 민감한 개인 정보를 포함할 수 있다. 따라서 데이터를 보호하기 위한 적절한 보안 기법을 필요로 한다.

스마트 시티는 다양한 데이터를 여러 센서 인프라를 통해 수집하고 자원, 도시 환경 등을 관리하는 도시이다. 예를 들어, 교통량 데이터를 수집하고 그 데이터에 기반을 두어 신호 체계를 유기적으로 조절할 수 있다. 이러한 스마트 시티에 수집되는 정보의 무결성을 보장할 수 없다면 수집된 데이터를 활용할 수 없기 때문에 데이터의 무결성을 지키고 동시에 투명하게 운영할 기법이 필요하다.

이러한 기존 스마트 홈이나 스마트 시티의 보안 기법은 클라우드, 사용자 애플리케이션 등 중앙의 관리 주체에 의한 접근 제어 형태를 가진다. 중앙 관리 주체에 의한 접근 제어는 중앙 관리 주체의 권한을 탈취하는 것만으로 데이터를 조작하거나 삭제, 탈취할 수 있다. 따라서 최근에는 기존 보안 기법보다 강력한 보안 기법의 필요성이 대두되고 있다.

이러한 문제의 해결책으로서 안전한 IoT 관리를 위해 블록체인 기술을 접목하여 연구가 진행되고 있다. [5]의 논문에서는 기존의 스마트 홈 보안 기법의 문제를 해결하기 위해 접근 권한과 데이터를 블록체인에 기록하는 방법을 제안하다. 접근 권한과 데이터를 블록체인에 기록함으로써 데이터의 무결성을 확보한다. 또한 외부 공격자가 중앙 관리 주체 권한을 탈취하는 것만으로 접근 제어를 무력화할 수 없다. 그러나 블록체인의 합의 알고리즘이 가지는 한계로, 확장성과 처리 속도에 제약이 있다. 연구 [6]에서는 IoT 장치로 인해 생기는 취약점에 대비하기 위한 권한 등 데이터의 무결성을 위해 블록체인 기술을 활용한 사례이다. 그러나 합의 알고리즘의 한계로 인해 확장성과 연산속도에 제약이 있다.

스마트 시티 IoT에 관한 연구 중, [7]에서는 시티 내의 자동차, 물류 트럭과 같은 교통수단에 블록체인을 접목하는 시스템을 제안했다. 제 3 신뢰 기관 없이 적은 투자로 데이터의 무결성을 유지하고 데이터를 공개하면서도 자료의 신뢰성을 확보할 수 있음을 목표로 한다. [8]에서는 스마트 시티에서 연결이 불안정한 IoT 노드로 인해 블록체인에서 블록이 전파되지 않는 문제의 해결법을 제시했다. 그러나 블록체인을 실제 모델에 적용하기에는 [10]에서 제시한 블록체인의 확장성으로 인한 블록 전파 문제와 연산 속도에 대한 문제가 남아있다.

2. 블록체인과 합의 알고리즘

블록체인은 제 3 신뢰 기관 즉 중계자의 도움 없이 데이터의 무결성을 제공하기 위해 제안된 분산 장부 시스템이다. 블록체인에서 데이터는 블록에 기록하며, 블록의 해시 값을 다음 블록에 기록하고 연결함으로써 데이터의 무결성을 확보한다. 그림 1은 비트코인의 블록 구조이다. 이렇게 연결된 블록은 중간의 블록이 변조된 경우 연결된 두 블록에 기록된 해시값이 달라지기 때문에 데이터 변조 여부를 확인할 수 있다. 모든 참여자가 동일한 블록 상태를 가지기 위해 합의 알고리즘을 사용한다[11]. 대표적으로 PoW (Proof of Work, 작업증명), PoS (Proof of Stake, 지분증명), DPoS 등의 합의 알고리즘을 활용하고 있다.

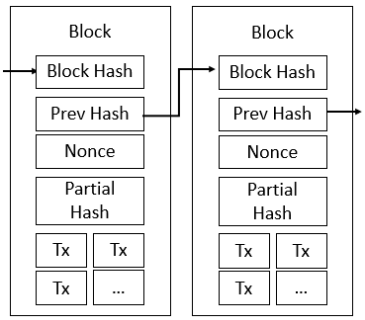

Fig. 1. Block Structure of Bitcoin.
그림 1. 비트코인의 블록 구조

PoW 합의 알고리즘은 해시 퍼즐 해결 속도에 따라 블록 결정 권한을 부여하는 연산력 기반의 합의 알고리즘이다. 비트코인[12] 등의 블록체인 플랫폼이 PoW 합의 알고리즘을 적용했다.

비트코인 이후 등장한 이더리움 등의 블록체인 시스템은 암호화폐를 기반으로 스마트 컨트랙트 실행을 위해 트랜잭션에 데이터를 포함할 수 있다[13]. 이후 합의 알고리즘을 개선한 EoS 등의 여러 블록체인 시스템이 등장했다.

이후 암호화폐의 지분에 따라 블록 증명에 기여함으로 연산력의 낭비를 줄일 수 있는 PoS가 등장했다. 그러나 모든 노드가 블록 증명에 기여함으로 인해 트랜잭션 속도가 느려지거나 악의적 노드로 인해 분기가 해소되지 않고 트랜잭션의 무결성이 보장되지 않을 가능성이 있다[14].

DPoS는 모든 노드가 블록 생성에 참여해서 생기는 확장성 문제를 해소하고 악의적 사용자의 블록 생성 검증 절차를 막기 위해 일정 수준 이상의 지분을 위임받은 노드에게 위임하는 합의 알고리즘이다. 지분을 위임받아 블록을 생성하는 역할을 맡은 노드를 BP(Block Producer)라고 한다. 즉, 분산성을 희생하고 확장성을 확보한 체계이다[15]. 본 논문에서는 EOSIO 기반으로 구현된 DPoS 방식의 블록체인을 통해 시스템을 구축하였다.

IoT 장치는 블록체인에 직접 참여할 만한 충분한

자원을 가지고 있지 않다. PoS 합의 알고리즘에서 블록을 생성할 대표 노드를 정하고 대표 노드간의 합의로 블록을 생성하는 방식인 DPoS를 기반 체계를 구현함으로써 기존 블록체인을 활용한 IoT 관리 프레임워크의 약점인 확장성과 처리 속도 손실을 최소화 하며 데이터 무결성을 확보할 수 있다.

구현 시스템은 EOSIO를 기반으로 설계한 DPoS 블록체인, IoT기기(스마트 조명, 라즈베리파이 기반 기기), 그리고 DApp으로 구성한다. IoT 프레임워크를 위해 데이터의 무결성과 함께 다량의 트랜잭션을 처리할 수 있는 블록체인 체계를 구현하기 위해 DPoS 기반 블록체인을 구성하였다. 명령을 읽어오거나 입력하는 동작을 모두 진행할 수 있는 스마트 조명과 센서기반 IoT기기를 블록체인 네트워크에서 동작하도록 구성하였다.

III. 기반 되는 시스템 구조

본 장에서는 기반 되는 시스템의 구조를 설명한다. 이 구조는 기존 연구[9]에서 제안된 내용으로서 다양한 IoT 기기들을 안전하게 낮은 오버헤드로 관리하기 위해 DPoS 합의 알고리즘을 사용하여 블록체인 네트워크를 구성하였다. 그림 2는 이러한 구조를 빌딩 관리 예시로 도식화하였다. 제안된 시스템 구조는 블록체인 네트워크(Blockchain Network)와 스마트 컨트랙트(Smart Contract), 게이트웨이(Gateway), 단말 애플리케이션(DApps, Decentralized Applications), IoT 장치(IoT Devices)로 구성된다.

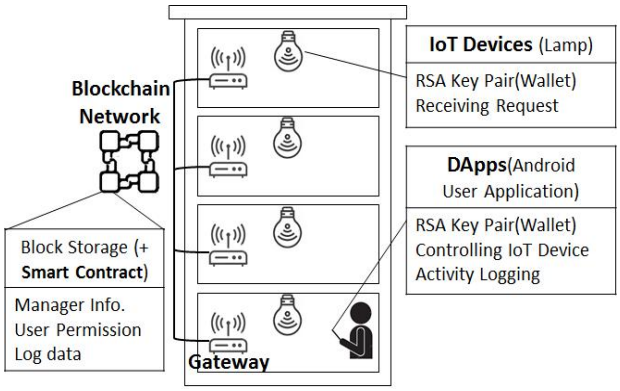

Fig. 2. Based System Architecture of Blockchain based IoT Management(E.g., Building Management)

그림 2. 기반 되는 블록체인 기반 IoT 관리 시스템 구조도 (빌딩 관리 예시)

게이트웨이는 사용자와 IoT 장치를 연결하고 블록체인 서비스 활용을 위한 API 엔드포인트를 제공하며 블록을 저장하기 위한 저장소를 포함한다. 스마트 컨트랙트는 블록체인을 IoT 접근제어 등으로 활용하기 위해 트랜잭션을 처리하기 위한 전자 계약서이다. IoT 장치는 단말 애플리케이션의 요청을 수신하고 실제 요청을 수행하는 장치이다. 단말 애플리케이션은 스마트폰에 설치되어 IoT 장치에게 요청을 전달하고 블록체인에 기록을 남기는 애플리케이션이다.

구성된 환경에서 IoT 장치의 활동 기록을 블록체인에 저장함으로써 기존 중앙 DB에 저장하는 시스템과 비교해서 얻을 수 있는 장점은 다음과 같다. 첫째, 기록을 블록체인에 저장함으로 블록체인의 특성인 불변성으로 블록체인이 유지되는 한 기록을 영구적으로 보존할 수 있다. 추가로 기록의 암호화를 통해 맞는 키를 보유한 사용자만 기록을 읽을 수 있도록 구조를 작성한 경우 개인키 또한 보존되어야 한다. 다만 이와 같은 암호화를 통해 기록은 읽을 수 없더라도 트랜잭션의 패턴을 분석하는 방법 등을 통해 사용자를 특정할 가능성이 있다. 이를 방지하기 위한 더미 트랜잭션 생성 등의 방법 또한 함께 사용할 수 있다.

둘째, 데이터를 서로 다른 응용에서 활용하기 적합하다. 특정 플랫폼이 아닌 블록체인의 데이터 구조에 따라 데이터가 정형화되므로 다른 응용에서 데이터를 활용하기 용이하다.

셋째, 개인정보 노출의 가능성이 적어진다. 블록체인 기반 기기 관리와 달리, 플랫폼에 종속된 장치가 해당 플랫폼에서 관리되는 경우(예, 구글플랫폼과 구글AI스피커), 생성되는 자료는 사용자의 동의하에 관리정보가 플랫폼에 저장되어 노출 가능성이 있다. 이러한 정보는 특정 사용자에게 맞춤형 서비스를 제공하거나 데이터를 분석하기 위해 저장되지만 프라이버시에 민감한 정보일 수 있다. 그러나 블록체인에 저장되는 경우 사용자 구분을 사용자 인식이 쉬운 정보(예, 이메일주소)로 하는 것이 아니라 인식이 어려운 정보(예, RSA 기반 공개키)를 사용하므로 블록에 저장된 정보가 노출되더라도 개인을 인식하기 어려워져 개인정보 노출 위협을 줄이고 기기 관리를 할 수 있다.

Ⅳ. 제안 시스템 구현 방안

1. 구현 시스템의 개요도

제안하는 시스템 프레임워크는 DPoS 기반의 EOSIO[16]를 기반으로 구현하였다. 구현된 시스템의 개요도는 그림 3과 같다. 구현파트를 블록체인 API, 사용자DApp, 컨트랙트, 장치로 나누어 설명한다.

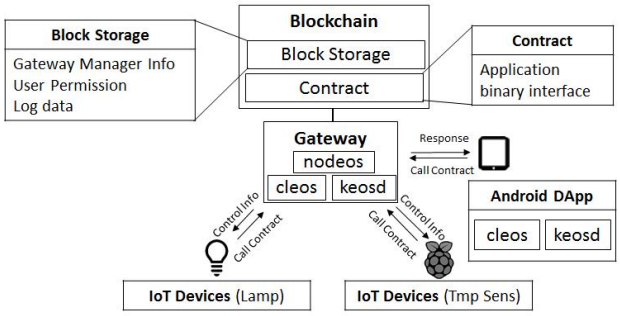

Fig. 3. Implemented System Architecture
그림 3. 구현된 시스템의 개요도

1) 블록체인API

블록체인과 소통하기 위한 API 부분이다.

- **nodeos** : v1.4.1을 사용하며 블록을 생성하고 블록체인 API의 접근 지점을 제공하는 핵심 데몬이다.
- **cleos** : nodeos 및 keosd와 상호작용하는 명령어 인터페이스이다.

2) 사용자 DApp

DApps에서 사용자를 구분하기 위한 계정과 인증하기 위한 키를 관리한다.

- **keosd(지갑)** : EOSIO의 RSA 키 쌍을 보관하고 서명을 위한 지갑 관리 에이전트이다.
- **계정** : 블록체인에 등록된 계정으로 알파벳과 숫자 1~5로만 이루어져 있으며 권한 조정을 위해 소유권 키와 활동 키가 최소 각 한 쌍씩 필요하다.
- **키** : keosd를 통해 발급된 RSA 키 쌍이다. 필요한 권한에 따라 저장할 수 있다.

3) 컨트랙트

컨트랙트 개발 및 배포에 필요한 요소이다.

- **EOSIO.cdt** : v1.4.1을 사용하며 C++ 기반으로 작성된 코드를 eosio-cpp를 통해 웹 어셈블리(wasm) 파일로 ABI 파일을 생성한다.
- **스마트 컨트랙트** : 계약 조건이 맞으면 자동으로 실행되는 계약서를 의미한다. 웹 어셈블리 형식에 따라 계정에 발행되고 매개변수를 담아 호출하여 사용한다.
- **ABI**(Application binary interface) : 컨트랙트에서 데이터를 주고받는 규칙에 대한 JSON과 Binary 명세서이다.

4) 장치

블록체인 요소를 포함하는 물리적인 장치이다. 네트워크를 구성하고 DApp을 통해 관리, 운용되는 물리적인 장치들이다. 사용자 요청에 따라 트랜잭션을 생성하고 요청을 수행한다. 각 장치들은 무선 네트워크를 통해 게이트웨이와 연결되어 있다.

- **스마트 조명** : LAN Control을 활성화 한 XIAOMI Yeelight Bedside lamp 2를 활용한다. 전원 제어, 밝기 조절, 색상 조절 등을 지원한다.
- **게이트웨이** : Ubuntu 18.04 LTS 환경에서 동작하고 EOSIO 1.4.1 및 EOSIO.cdt 1.4.1를 갖춘 Nodeos를 실제로 동작하는 장치이다. 블록체인에 접근할 수 있는 주소를 제공한다.
- **온습도 감지 센서 IoT기기** : 라즈베리파이 3 B+ 모델에 Rasbian 을 통해 연결한 DHT11 온습도 감지 센서를 사용한다.
- **DApp기기** : IoT 장치를 제어하기 위해 같은 게이트웨이에 연결되어 있는 장치이다. Java 기반의 안드로이드 애플케이션이다. 네트워크와 사용자를 생성하고 IoT 장치를 조작하고 해당 내용을 블록체인에 기록한다.

2. 구현 시스템의 프로세스 및 운용

구성한 시스템 구조에서 처리 프로세스를 설명하기 위해 그림 4에 시퀀스 다이어그램으로 도식화하였다. 사용자와 장치를 등록하고, 권한에 맞추어 명령을 내리는 절차이다.

각 절차는 사용자 등록(Seq1), 장치 등록(Seq2), 장치 제어 기록 절차(Seq3)이다. 사용자 등록 절차(Seq1)는 애플리케이션에서 Keosd에서 제공하는 키 발급 절차를 통해 습득한 공개키를 블록체인에 등록해서 권한을 할당하는 절차이다. 네트워크 접근을 허가하고자 하는 사용자 이름을 담은 컨트랙트 생성하고 네트워크 관리자의 서명을 담은 트랜

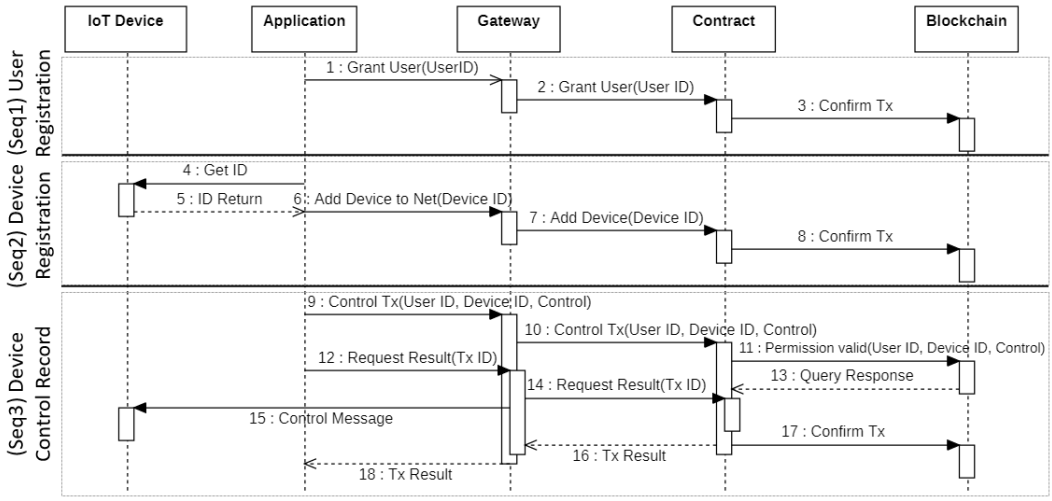

Fig. 4. Sequence Diagram of Implemented System.
그림 4. 구현된 시스템의 시퀀스 다이어그램

잭션을 블록체인에 전달한다. 장치 등록(Seq2)은 장치의 ID를 요청한 게이트웨이에 가입하는 과정이다. 장치를 구분할 수 있는 ID를 장치에게 얻어온 트랜잭션을 생성하고 네트워크 관리자의 서명을 담아 블록체인에 전달한다. 장치에게 기록을 남기며 전달하는 과정(Seq3)은 앞서 진행한 절차를 통해 정당한 권한을 얻은 사용자의 서명과 장치 ID를 담은 명령을 블록체인에 전달한다. 정당한 요청인 경우 블록체인에 기록이 남고 게이트웨이는 IoT 장치에게 명령을 전달한다. 제안 시스템을 구현함으로써 트랜잭션을 분산 네트워크인 블록체인에 저장하여, 투명하고 영구적이며 추적 가능한 관리 프레임워크를 만들 수 있다.

컨트랙트는 eosio.cdt로 컴파일 할 수 있는 C++ 기반 언어로 작성하였다(그림 5 참조). 작성한 컨트랙트 코드는 eosio.cdt로 컴파일 한다. 그 결과로 wasm(WebAssembly)[17]와 JSON 기반의 ABI[18] 파일을 생성한다(그림 6 참조). 생성한 wasm 파일과 ABI파일을 블록체인에 전송한다. 확인하는 서명을 통해 컨트랙트를 등록한 계정이 컨트랙트의 메소드를 수행할 준비를 마친다.

```
[[eosio::action]]
void adduser(name contractcreator,name wantuser) {
  require_auth(contractcreator); //Require contractcreator's Sign
  if(checkdeployer(contractcreator) == 0) return;
  puser_index ulist(_code, _code.value);
  auto iterator = ulist.find(wantuser.value);
  if( iterator == ulist.end() )//add user data to Blockchain
  {
    ulist.emplace(wantuser, [&]( auto& row ) {
      row.user = wantuser;
    });
  }
}
```

Fig. 5. An Example of Source Code based on C++.
그림 5. C++ 기반으로 작성된 소스코드 예

```
"version": "eosio::abi/1.1",
"structs": [
    {
        "name": "adduser",
        "base": "",
        "fields": [
            {
                "name": "contractcreator",
                "type": "name"
            },
            {
                "name": "wantuser",
                "type": "name"
            }
        ]
    },
```

Fig. 6. An Example of ABI Code based on JSON.
그림 6. JSON 기반으로 작성된 컨트랙트의 ABI 예

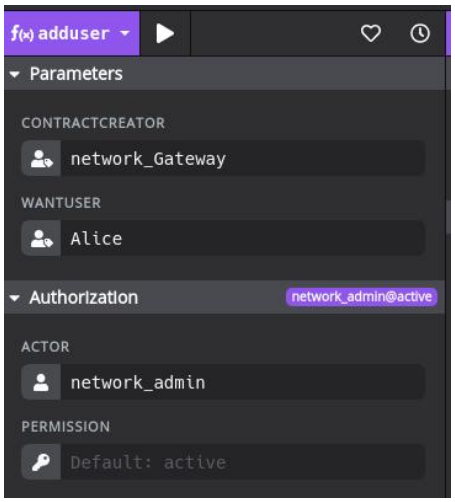

Fig. 7. An example of Executable Contract based on ABI and WASM.
그림 7. ABI와 WASM 파일을 통해 작성된 컨트랙트 예

그림 7은 작성된 컨트랙트를 EOS Studio[19]의 시각화 도구를 통해 보여진 실행 가능한 컨트랙트의 모습이다.

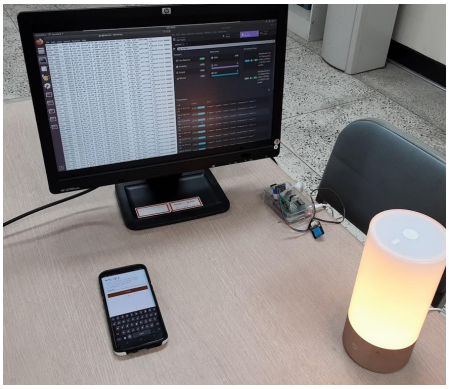

Fig. 8. Experiment Screenshot of Implemented System.
그림 8. 구현된 시스템의 실험 화면

그림 8은 구현된 시스템의 실험 화면이다. 실제 구현 후 실험 환경을 구축한 후 블록체인에 키를 생성하고 해당키를 사용하는 관리자 계정을 생성하였다. 계정에 사용자 등록, 장치 등록, 장치 제어 기능을 가진 컨트랙트를 배치하고, 네트워크를 사용할 사용자 계정과 키를 생성하여 사용자 등록 절차를 진행하였다. 네트워크에 등록할 장치를 선택하고 장치 등록 절차를 진행하였다. 등록된 사용자는 등록된 장치를 대상으로 조작 요청을 전달한다. 그림 9와 10은 DApp을 통해 장치 등록과 제어된 장치(스마트램프)의 화면을 보여주고 있다.

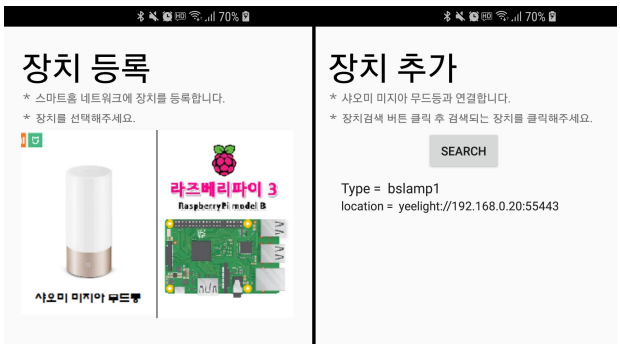

Fig. 9. DApp Screenshot for Device Registration.
그림 9. DApp에서 장치 등록 화면

V. 구현 결과와 성능 평가

본 장에서는 구현한 시스템을 유사 타 시스템과 비교하여 본 시스템의 이점을 분석하고, 구현된 시스템의 수행 시간을 측정하여 시간 측면의 오버헤드를 분석하고자 한다.

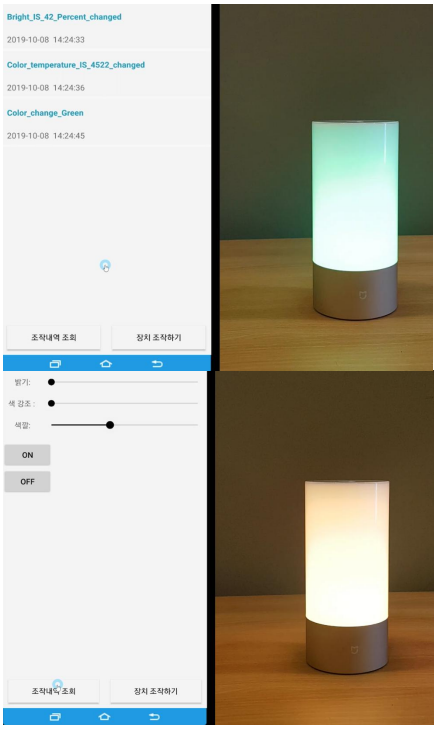

Fig. 10. Screenshot of Controled Device on DApp.
그림 10. DApp에서 제어된 장치와 화면

1. 타 시스템과 비교

표 1은 IoT 관리를 위한 유사 타 시스템의 비교 내용을 정리한 것이다. 표에서 보이는 것처럼 4개의 시스템은 장치 관리, 특히 권한 관리를 위해 구축되었다. 그러나 블록체인 기반 관리 시스템들에서만 해당 권한 제어 활동 기록들이 블록체인 기술에 의해 준 영구적으로 관리될 수 있다. 특히, 현재 운영되고 있는 IBM Watson과 젬알토 Safenet은 고성능 장치 또는 비용이 요구되거나 특정 목적의 응용을 위해 고안된 시스템이다. 블록체인 기술을 활용한 본 시스템의 장점으로서 앞서 소개한 대로 데이터를 서로 다른 응용에서 활용하기 적합하고, 개인정보 노출 가능성도 적어진다는 장점도 있다.

2. 시간 오버헤드 분석

본 절에서는 제시한 블록체인 기술의 도입으로 인한 시간 측면의 오버헤드를 측정한다. 인증 등의 절차의 변수를 제거하고, 블록체인의 혼잡 여부에 따라 IoT 장치 조작 후 블록기록 요청의 전송 시간(그림 4의 9~15번 과정)을 측정한다. 즉, IoT 장치

Table 1. Comparison of other systems.
표 1. 타 시스템과 비교

Factors	Our system	IBM Watson [20]	Gemalto Safenet [21]	Google Cloud IoT [22]
Device rights management	O	O	O	O
Semi-permanent activity record management	O	O	O	X
No high performance device or cost	O	X	X	X
Goal	IoT-based system management and malicious behavior monitoring	Logistics, equipment realtime monitoring	Power trading, Blockchain system for consumption monitoring	Cloud service for managing Android Things
Usage system	Blockchain	Blockchain	Blockchain	Cloud Server

를 제어한 후 블록체인에 트랜잭션을 제출하고 트랜잭션 정보를 받아오는 시간이다. 이 때 사용한 IoT 장치는 XIAOMI의 조명 장치 Yeelight Bedside lamp이고 이를 무선랜을 통해 제어한다. 블록체인 환경은 Idle상태와 많은 트랜잭션이 동작하는 Busy 상태에서 비교하였다. Busy상태는 다른 사용자가 장치를 조작하는 많은 트랜잭션을 입력하는 상황으로, 한 블록 당 30개에서 38개 정도의 트랜잭션이 들어가고, 초당 2개의 블록이 생성되는 상황이다. 블록체인을 사용하지 않는 경우(WithoutBC)에는 게이트웨이에 파일로 저장하는 방식이다. 그림 11의 결과는 각 상황에 대해 30개의 트랜잭션에 대해 측정하고, 한 트랜잭션 당 시간을 초로 나타내

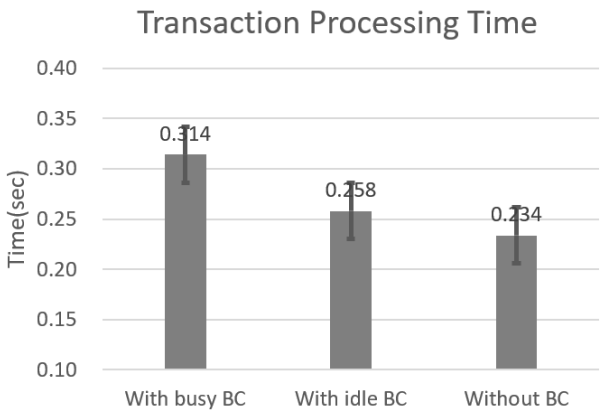

Fig. 11. Transaction processing time.
그림 11. 트랜잭션 처리 시간

었다.

결과, 블록체인 기반 Idle상태와 서버 방식의 WithoutBC와 큰 차이가 없음을 볼 수 있다. 다만 트랜잭션이 블록체인에 실시간으로 등록되어야 제안한 체계가 동작할 수 있는 만큼, 혼잡한 Busy상태의 블록체인인 경우 다소 지연이 발생함을 보여주고 있다. 그러나 제안한 구현 시스템에서 요구하는 트랜잭션의 연산은 높은 부하를 요구하지 않기 때문에 블록체인 네트워크가 혼잡한 경우에서도 적절한 트랜잭션 속도를 보장 받을 수 있다.

서버가 아닌 블록체인에 사용자와 장치를 등록하고 제안 시스템에서 실제로 장치가 등록됨을 확인하기 위해서는 각 블록이 블록체인에서 확인할 수 있도록 확정되어야 한다. 그림 4의 시퀀스 다이어그램의 1~3번 과정을 Seq 1.사용자 등록(User Registration), 4~8번 과정을 Seq 2.장치 등록(Device Registration), 9~18번 과정을 Seq 3.장치 제어기록(Device Control Record)으로 한다. 각 시퀀스마다 블록체인에 접근하고 블록이 확정되어 확인하는 시간을 측정하여 블록체인을 사용함으로 인해 생기는 오버헤드를 측정한다.

그림 12는 이러한 블록 확정 시간에 대해 트랜잭션을 30회 반복 수행하여, 각 시퀀스(Seq1, Seq2, Seq3) 수행시간을 측정하고 평균과 표준 편차를 표기하였다. 실험 시, EOSIO의 블록 생성 주기가 0.5초이므로 같은 시퀀스임에도 오차가 발생할 수 있다.

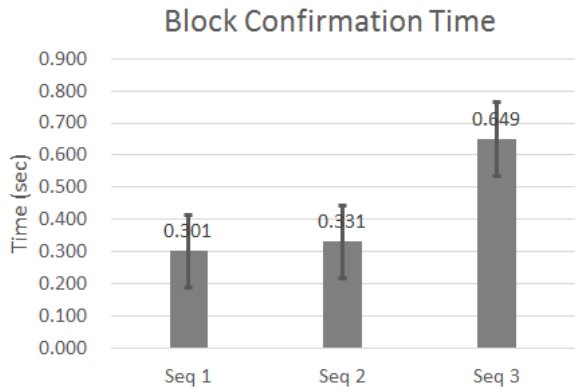

Fig. 12. Block Confirmation time.
그림 12. 블록 확정 시간

실험 결과, Seq2가 Seq1보다 장치와의 인터렉션 때문에 시간이 약간 더 걸렸고, Seq3은 Seq1/Seq2보다 2배정도의 시간이 걸렸음을 알 수 있다. 또한 실험 결과는 트랜잭션을 실제로 수행한 시간에 따

라 약간 영향을 받을 수 있지만, 일관된 편차를 가짐을 알 수 있다.

Ⅵ. 결론

본 논문에서는 선행 연구에서 제안한 DPoS 합의 알고리즘을 통해 실행속도와 확장성을 얻은 블록체인 기반 IoT 관리 구조를 실제로 구현하고 그 성능을 분석하였다. 구현한 시스템에는 사용자 DApps을 자바 기반 모바일 애플리케이션으로 구현하고, 실제 IoT 기기(라즈베리파이 기반 기기, 스마트 램프)를 포함하여 DApps을 통해 제어되며, 장치제어 기록은 블록체인에 기록되도록 구현하였다. 사용자 입장에서 실제 IoT 기기를 DApp으로 동작시키고 블록체인을 통해 관리하는 이점과 시간측면의 오버헤드를 분석하여 본 시스템의 실현 가능성을 보였다.

References

[1] “IoT devices grow 2.5x in 2022, expecting half the world's networking devices,” Science Times, 2019.

[2] Lucas Mearian, “IoT can be a blockchain killer app… Active PoC of large companies,” IT World, 2018.

[3] B. L. Risteska Stojkoska and K. V. Trivodaliev, “A review of Internet of Things for smart home: Challenges and solutions,” *Journal of Cleaner Production,* vol.140, pp.1454-1464, 2017.
DOI: 10.1016/j.jclepro.2016.10.006

[4] Y. Seo, J. Song, Y. Kong, “Blockchain Technology: Prospect and Implications in Perspective of Industry and Society,” SPRI Issue report, No.2017-004, 2017.

[5] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, “Blockchain for IoT security and privacy: The case study of a smart home,” *in Proc. of IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops),* pp.618-623, 2017.
DOI: 10.1109/PERCOMW.2017.7917634

[6] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, “Smart Contract-Based Access Control for the Internet of Things,” arXiv: 1802.04410 [cs], 2018.
DOI: 10.1109/JIOT.2018.2847705

[7] P. Manjunath, R. Soman and D. P. Gajkumar Shah, “IoT and Block Chain driven Intelligent Transportation System,” *2018 Second International Conference on Green computing and Internet of Things (ICGCIoT),* pp.290-293, 2018.
DOI: 10.1109/ICGCIoT.2018.8753007

[8] S. Kushch and F. Prieto-Castrillo, “Blockchain for Dynamic Nodes in a Smart City,” *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pp.29-34, 2019.
DOI: 10.1109/WF-IoT.2019.8767336

[9] M. Kim, Y. Kim, “Development of IoT Device Management System Using Blockchain DPoS Consensus Algorithm,” *Journal of IKEEE,* vol.23, no.2, pp.508-516, 2019.
DOI: 10.7471/ikeee.2019.23.2.508

[10] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, “A survey on the security of blockchain systems,” *Future Generation Computer Systems,* p. S0167739X17318332, 2017.
DOI: 10.1016/j.future.2017.08.020

[11] G.-T. Nguyen, and K. Kim, “A Survey about Consensus Algorithms Used in Blockchain,” *Journal of Information Processing Systems,* vol.14, No.1, pp.101-128, 2018.
DOI: 10.3745/JIPS.01.0024

[12] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008, https://bitcoin.org/bitcoin.pdf. 2019.

[13] “Introduction to Smart Contracts — Solidity 0.5.8 documentation.” https://solidity.readthedocs.io/en/v0.5.8/introduction-to-smart-contracts.html, 2019.

[14] V. Buterin, “On Stake,” 2014. https://blog.ethereum.org/2014/07/05/stake/.

[15] B. Xu, D. Luthra, Z. Cole, and N. Blakely, “EOS: An Architectural, Performance, and Economic Analysis,” https://whiteblock.io/library/eos-test-report.pdf.

[16] EOSIO, https://github.com/eosio.
[17] “WebAssembly,” [Online]. Available: https://webassembly.org/.
[18] “application binary interface - an overview | ScienceDirect Topics,” [Online]. Available: https://www.sciencedirect.com/topics/computer-science/application-binary-interface.
[19] “EOS Studio – Graphic IDE for EOSIO Development,” [Online]. Available: https://www.eosstudio.io/.
[20] “Blockchain Security Solutions | Bring Trust to Blockchain with Gemalto,” Gemalto. Available: https://safenet.gemalto.com/blockchain/.
[21] “Explore the Internet of Things (IoT),” 05-Dec-2019. Available: https://www.ibm.com/internet-of-things.
[22] “Google Cloud IoT - Fully managed IoT services,” Google Cloud. [Online]. Available: https://cloud.google.com/solutions/iot/.

BIOGRAPHY

Mihui Kim (Member)

1997 : B.S. in Dept. of Computer Science and Engineering, Ewha Womans University, Korea
1999 : M.S. in Dept. of Computer Science and Engineering, Ewha Womans University, Korea
1999~2003 : Researcher, ETRI(Electronics and Telecommunication Research Institute), Korea
2007 : Ph.D in Dept. of Computer Science and Engineering, Ewha Womans University, Korea
2007~2009 : Full Time Lecturer, Dept. of Computer Science and Engineering, Ewha Womans University, Korea
2009~2010 : Postdoctoral Researcher, Computer Science, North Carolina State University, USA
2011~Current : Associate Professor, Dept. of Computer Science & Engineering, Hankyong National University, Korea
Research interests : Security and efficient protocol design in IoT and crowdsensing system, Blockchain technologies

Youngmin Kim (Member)

2013~Current : MS student, Dept. of Computer Science & Engineering, Hankyong National University, Korea
Research interests : Security in IoT and crowdsensing system, Blockchain technologies