Journal of KIIT. Vol. 20, No. 12, pp. 103-114, Dec. 31, 2022. pISSN 1598-8619. eISSN 2093-7571 
http://dx.doi.org/10.14801/jkiit.2022.20.12.103

# 웹 어셈블리 활용한 웹 기반 블록체인 네트워크 시스템 설계 및 구현

송우석*, 정현준**[1], 정동원**[2]

# Web-based Blockchain Network System Design and Implementation using Web Assembly

Wooseok Song*, Hyunjun Jung**[1], and Dongwon Jeong**[2]

이 논문은 정부(과학기술정보통신부)의 재원으로 한국 연구재단의 지원을 받아 수행된 연구임 (No. NRF-2022R1G1A1008493), 본 연구는 환경부/한국환경산업기술원 지중환경오염위해관리기술개발사업 (2022002450002)으로 수행되고 있습니다.

## 요 약

본 논문에서는 웹 환경에서의 자바스크립트와 웹 어셈블리 간 블록체인의 블록 생성 성능을 비교 측정하고 웹 환경에서 블록체인 시스템을 유지하는 방법을 제안한다. 최근 블록체인의 규모가 증가하는 가운데 현재 대부분의 블록체인 네트워크에 참여하기 위해서는 별도의 클라이언트를 설치해야 한다. 이는 블록체인 기술의 대중화를 위한 편의성을 저해할 수 있는 요인이다. 따라서 이 논문에서는 사용자들에게 높은 접근성을 제공할 수 있도록 웹 기반의 블록체인 시스템을 제안한다. 실험 결과 웹 어셈블리가 자바스크립트 대비 블록 생성 측면에서 약 57.62% 우수한 성능을 보였다. 또한, 제안한 시스템을 통해 사용자가 브라우저를 통해 블록체인 네트워크에 참여하고 블록을 생성 및 검증하며 네트워크를 유지할 수 있음을 보였다.

## Abstract

This paper measure the block generation performance of blockchain between javascript and web assembly in web enviroment and propose a method to maintain a blockchain system in a web environment. As the recent growth of blockchain, it is necessary to install a client software to participate in most blockchain networks. This is a factor that can hinder the convenience for popularization of blockchain technology. Therefore, this paper proposes a web-based blockchain system to provide users with high accessibility. In the experiment, the web assembly showed about 57.62% better performance than javascript in terms of blockchain block generation in the web environment. In addition, the proposed system shows that users can participate in the blockchain network through a browser and maintain the network by create and verify blocks.

Keywords

blockchain, web assembly, javascript, native, WebRTC, P2P

* 군산대학교 소프트웨어학과 학사과정
- ORCID: https://orcid.rog/0000-0002-4574-9280
** 군산대학교 소프트웨어학과 교수(교신저자)
- ORCID[1]: https://orcid.org/0000-0002-6717-1395
- ORCID[2]: https://orcid.org/0000-0001-9887-5336

· Received: Oct. 19, 2022, Revised: Dec. 06, 2022, Accepted: Dec. 09, 2022
· Corresponding Author: Hyunjun Jung and Dongwon Jeong
Dept. of Software at Kunsan National University, 558, Daehak-ro, Gunsan, Jeollabuk-do, Republic of Korea
Tel.: +82-63-469-8917, Email: {junghj85, djeong}@kunsan.ac.kr

# Ⅰ. 서 론

최근 웹과 블록체인 동향을 보았을 때 그 규모는 지속해서 커지고 있다. 블록체인은 빅데이터, 인공지능, 로봇공학, 사물인터넷과 더불어 4차 산업혁명과 웹 3.0을 이끌어갈 핵심 기술로서 주목을 받고 있다[1]. 블록체인은 데이터의 투명성과 무결성을 보장하는 분산 원장 저장 기술로 사토시 나카모토에 의해 제안되었다[2].

이 연구에서는 전자화폐에 대한 개념과 이를 P2P(Peer-to-Peer) 방식으로 운영하는 방법에 대해 처음 설명하였다. 사용자는 블록체인을 통해 네트워크 상에서 발생하는 데이터와 정보를 기록 혹은 조회할 수 있다. 블록체인 네트워크에 기록된 데이터들은 하나의 블록 단위로 수집 및 저장된다.

최근에는 PWA(Progressive Web Application)등 웹 기술을 이용하여 네이티브 프로그래밍 환경에서 구동하던 애플리케이션들을 웹 앱과 하이브리드 앱으로 바뀌는 추세이다[3]. 기존의 네이티브 앱은 클라이언트를 내려받거나, 직접 PC에 설치해야 하는 번거로운 과정을 거쳐야 한다. 하지만 웹 앱은 PC와 모바일 등 브라우저가 설치된 모든 플랫폼에서 서비스할 수 있어 접근성이 높은 장점을 지닌다. 따라서 웹은 플랫폼에 비종속적이며 항상 인터넷의 중심에 있어 기존의 많은 네이티브 시스템을 웹에 연결 및 이식하려는 연구들이 지속해서 진행되어 왔다[4].

웹 애플리케이션에서 DOM(Document Object Model) 조작, 사용자 이벤트 처리, 웹 서버와의 통신 등 웹 클라이언트에서 필요한 기능적인 부분은 자바스크립트를 통해 구현된다. 하지만 자바스크립트는 많은 수학 연산을 요구하는 경우 네이티브 프로그래밍 언어에 처리 속도가 저하되는 한계를 지닌다[5]. 또한, 자바스크립트는 동적 타입의 언어로서 자바스크립트 엔진은 코드를 실행하기 전 해당 변수의 타입을 예측하는 시간이 필요하며, 코드를 실행하는 과정에서 예측한 타입이 틀렸을 경우 재최적화하는 단계가 추가로 발생한다. 이로 인해 기존의 네이티브 앱에 비해 많은 수학적 연산이 필요한 경우 수행 성능이 떨어진다[6].

웹 어셈블리는 이러한 자바스크립트의 단점을 보완할 수 있는 새롭게 제시된 저수준의 바이트코드이다. 국제 웹 표준 관리 컨소시엄인 W3C(World Wide Web Consortium)에서 공식 권고한 HTML, CSS, 자바스크립트에 이은 4번째 웹 언어이다[7].

웹 어셈블리는 새로운 프로그래밍 언어가 아닌 기존의 C, C++, Rust와 같이 네이티브 프로그래밍 언어의 코드를 브라우저가 해석할 수 있는 바이너리 파일로 컴파일하는 기술이다. 현재 웹 어셈블리는 LLVM 툴체인을 통해 구조화된 이진 파일로 컴파일되며 컴파일 과정에서 최적화를 진행하기 때문에 웹 브라우저에서 빠르게 로드하여 네이티브와 가까운 속도로 실행할 수 있게 해준다[8].

따라서 게임, 동영상 편집기 등 기존의 네이티브 시스템을 웹으로 이식할 때 많은 연산이 필요한 부분을 웹 어셈블리로 구현한다면 자바스크립트보다 속도 적 이점이 있으며 기존의 코드를 재활용할 수 있고 기존 성능의 한계로 서버에서 처리하는 연산을 클라이언트에 옮길 수 있는 장점이 있다.

현재 블록체인은 금융 분야뿐만 아니라 물류, 의료 등 다양한 분야에 활용된다. 대표적으로 국내 코로나 19 예방접종 정보를 블록체인에 담아 백신 인증 서비스를 제공하고 있는 COOV[9]가 있다.

블록체인 시장의 급격한 성장 가운데 현재 블록체인 네트워크에 참여하기 위해서는 한정된 플랫폼에 별도의 클라이언트를 설치해야 하는 번거로움이 있다. 이는 블록체인 기술의 대중화를 위한 편의성을 저해할 수 있는 요인으로 작용한다[10].

이 논문에서는 웹에서의 효율적인 블록체인 작업증명을 위해 웹 어셈블리와 자바스크립트의 작업증명의 효율성을 분석하고 사용자들에게 높은 접근성을 제공하기 위해 웹 기반의 블록체인 네트워크 시스템을 제안한다.

이 논문의 구성은 다음과 같다. 제2장에서는 관련 연구를 분석한다. 제3장에서는 임의의 블록체인 시스템의 동작 과정과 웹 기반의 블록체인 동작 과정에 관해 기술한다. 제4장에서는 각 환경에서의 블록 생성 시간을 측정하고 웹 기반의 블록체인 네트워크 구현 및 시각화 결과를 기술한다. 마지막으로, 제5장에서는 결론 및 향후 연구를 서술한다.

## II. 관련 연구

### 2.1 웹 어셈블리를 활용한 암호화 응용 연구

웹에서의 웹 어셈블리를 활용하여 자바스크립트 대비 해시 암호화 연산 속도와 안전성을 증가시키기 위한 연구가 진행되어왔다[11]. 암호화 알고리즘을 구현할 때는 기능적 정확성만이 중요한 것이 아니다. 이와 더불어 해당 과정을 수행하며 정보가 유출되는 것을 방지하는 것도 중요하다. 현대의 자바스크립트 런타임은 just-in-time(JIT) 컴파일러, Garbage Collector(GC) 등이 포함된 복잡한 시스템이지만 timing side-channels에 의해서 자바스크립트 코드가 유출될 가능성이 있다[12]. 해당 논문에서는 이러한 문제점을 해결하기 위해서 웹 어셈블리 기반의 암호화 방식을 제안하였다. 기존의 SHA-256 해시 함수 등의 암호화 알고리즘을 자바스크립트와 웹 어셈블리로 구현하고 이에 대한 성능 비교 평가하였다. SHA-256 함수란 표준 해시 알고리즘인 SHA-2 계열 중 하나이다. 임의의 길이를 가진 메시지를 입력으로 256비트의 고정된 길이의 해시 값을 출력한다. 현재 많은 블록체인 시스템에서 SHA-2 계열 해시 함수를 채택하여 사용하고 있다[13]. 해당 논문에서는 웹 어셈블리로 구현한 SHA-256 해시 함수 모듈이 자바스크립트 대비 우수한 성능을 보였지만 해당 실험을 통해 네이티브 환경 대비 웹 어셈블리와 자바스크립트의 성능 차이를 보이기에는 한계가 있다.

앞서 기술한 연구와 함께 웹 어셈블리를 활용한 블록체인 채굴 시스템을 감지하기 위한 연구가 진행되었으나[14], 이 연구를 통해서 웹 기반의 블록체인 활용 가능성을 파악하기에는 한계가 있다.

### 2.2 WebRTC 응용 연구

WebRTC란 웹 브라우저 간 플러그인 없이 P2P 네트워크를 구축하고 각 노드 간 실시간 통신을 가능하게 하는 API 표준이다[15]. 최근 WebRTC를 이용하여 브라우저 간 P2P 네트워크를 구축하여 이를 다양한 분야에 활용하려는 연구가 지속되어 왔다.

[16]은 온라인 학습 효율 증대를 위해서 WebRTC 기반의 원격 협업 학습 플랫폼을 제안하였다. 별도의 소프트웨어 없이 웹을 통하여 서비스되기 때문에 다양한 단말기에서 손쉽게 접근 및 이용할 수 있으며 기존의 웹의 취약점인 사용자 단말 간 멀티미디어 상호 커뮤니케이션 기능을 제공한다. 웹 브라우저만으로 교수자와 학습자 간의 고품질의 단대단 및 다대단 상호 스트리밍 연결이 가능하여 원격 교육에 적합한 특징을 가진다.

[17]은 기존의 네이티브 플랫폼 기반 솔루션의 전유물로만 여겨졌던 화상회의 시스템을 WebRTC와 순수 웹 기술을 이용하여 웹 플랫폼으로 구현하였다. WebRTC를 통해 P2P 기반 통신 방식을 구축함으로서 시스템 운영에 필요한 비용 절감 효과를 얻을 수 있으며 이기종 단말과 상호연동이 가능한 스트리밍 구조를 제안하여 기존 시스템과의 높은 상호 연동성이 특징이다.

[18]은 UAV(Unmanned Aerial Vehicle)를 이용하여 대기 질 측정, 교통 관찰, 기상 관찰 등 도시 및 산업 지역을 관찰하기 위해서 WebRTC를 이용하였다. 해당 논문의 실험에서는 드론(Drone)에 Raspberry pi 4 Model B를 장착하고 WebRTC API를 통한 드론의 실시간 영상 및 센서 데이터 전송을 수행하였다. Raspberry Pi 4 Model B 프로세서에서 실행되는 WebRTC 애플리케이션이 높은 해상도의 영상을 빠른 속도로 실시간 스트리밍할 수 있음을 보였다.

이처럼 대부분의 연구에서 웹이라는 플랫폼 특성을 이용하여 높은 접근성 및 이용성을 보장하며 동시에 WebRTC를 이용하여 기존의 네이티브 시스템을 웹으로 이식하려는 특징을 보였다.

## III. 제안 실험 및 시스템

본 장에서는 임의의 블록체인 시스템의 블록 생성 과정, 임의의 블록체인 시스템을 네이티브 환경과 웹 어셈블리 환경, 자바스크립트 환경에서의 구현 방법, 제안하는 웹 기반의 블록체인 네트워크 설계 및 각 노드의 채굴과 검증 과정을 통해 웹 기반 블록체인의 동작 과정을 기술한다.

### 3.1 임의의 블록체인 시스템 블록 생성 과정

본 절에서는 본 논문에서 제안하는 임의의 블록체인 시스템의 블록 생성 과정을 설명한다.

제안한 블록체인 시스템은 초기 해시값과 데이터를 할당받아 Genesis Block을 생성한다. 이후 생성되는 블록들은 해당 블록의 데이터와 함께 이전 블록의 해시값을 포함하며 새로운 해시값을 통해 이전 블록들과 연결된다. 블록이 연결된다는 의미는 새로운 블록의 해시값이 이전 블록의 정보에 대한 해시값을 포함하여 기록하는 것을 의미한다[19].

블록체인은 블록 생성량을 조절하고 블록체인 네트워크의 안정화를 위하여 난이도를 통해 블록 생성 시간을 조절한다[20]. 작업 증명은 임의의 논스값을 대입하여 제시된 타겟 값보다 작은 블록 해시 결과값을 찾도록 반복하는 과정이다. 난이도가 높을수록 정답 논스값을 찾는데 많은 연산이 수행된다.

각 블록의 해시값은 작업 증명 과정을 통해 나오며 이 논문에서 구현한 임의의 블록체인 시스템의 블록 생성 과정은 그림 1과 같다. Algorithm 1은 블록 생성 과정에서 필요한 작업 증명 알고리즘이다.

Algorithm 1: 임의의 블록체인 시스템의 작업 증명 알고리즘

```
1. Input b = timestamp+data+index+previousHash
2. Input nonce = 0
3. Start proofOfWork:
4.         Repeat nonce++
5.         Until SHA256(b, nonce) < targetDifficulty
6. End
```

### 3.2 각 환경에 따른 블록체인 동작 과정

본 절에서는 구현한 임의의 블록체인 시스템이 각 환경에 따른 동작 과정 차이에 관하여 기술한다.

웹 환경의 블록체인 시스템을 구현하기 위해서 C++를 활용한 웹 어셈블리 기반 환경과 자바스크립트 기반의 환경을 구현하고 네이티브 환경의 블록체인 시스템을 구현하기 위해서 C++를 이용한다.

네이티브 환경은 작성한 C++ 코드가 G++ 컴파일러에 의해서 최적화 및 컴파일되며 CPU 아키텍처에 맞는 실행파일을 생성한다. 웹 어셈블리를 이용한 웹 환경은 작성한 C++ 코드가 Emscripten 컴파일러에 의해서 최적화 및 컴파일되며 브라우저 엔진이 해석할 수 있는 wasm 파일을 생성한다.

자바스크립트와 웹 어셈블리 환경 모두 웹 브라우저에 의해서 코드가 해석 및 실행되며 모든 환경에서 사전에 설정한 난이도에 맞게 블록을 생성한다. 그림 2는 구현한 임의의 블록체인 시스템이 각 실험 환경에 다른 구동 방식 차이를 보여준다.

### 3.3 웹 기반 블록체인 네트워크 설계

본 절에서는 웹 기반 블록체인 네트워크를 구축하기 위한 필요 기술 및 서버와 그 역할을 기술한다. 블록체인은 중앙 집중형 서버에 거래 기록을 보관하지 않고 P2P 네트워크를 이용하여 모든 네트워크 참여자들에게 거래 내역을 공유하고 이를 통해 데이터 위조 및 변조를 할 수 없도록 구현되어 있다[21].

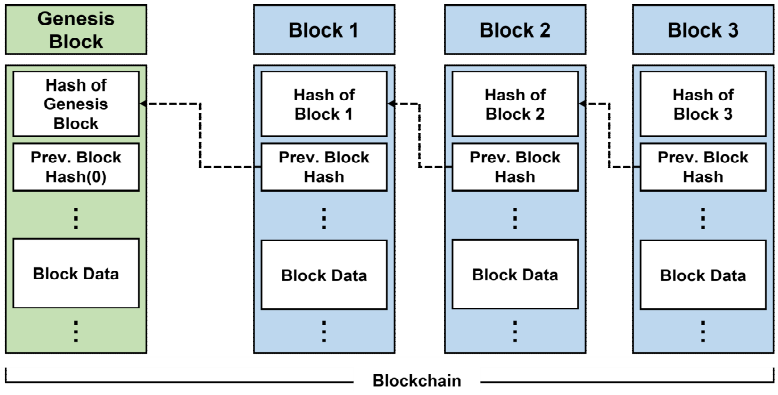


그림 1. 블록체인의 블록 생성 과정
Fig. 1. Block generation process in Blockchain

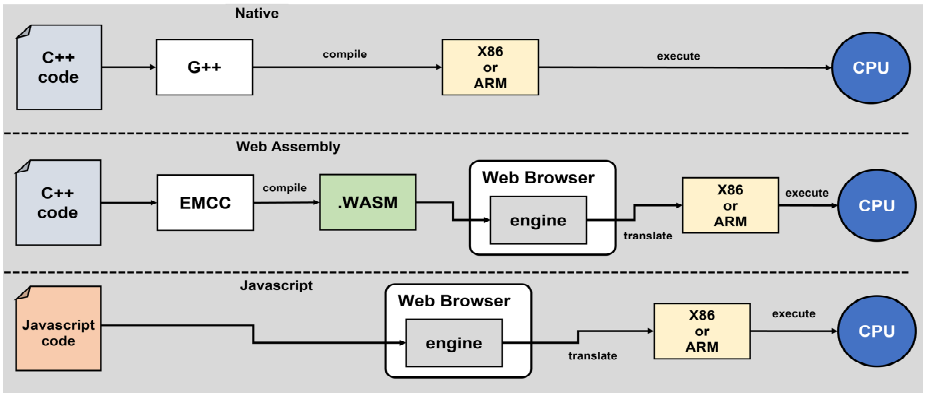


그림 2. 각 환경에 따른 블록 생성 수행 과정 차이
Fig. 2. Differences in the block creation process according to each environment

이 논문에서는 사용자가 웹 브라우저를 이용하여 P2P 네트워크에 참여하고 블록체인 네트워크를 유지하기 위해서 WebRTC(Web Real-Time Communication) API를 이용한다.

이 논문에서 제안하는 웹 기반의 P2P 블록체인 네트워크 시스템을 구현하기 위해서 시그널링 서버(Signaling server), 블록체인 네트워크 관리 서버, 웹 서버, 데이터 서버 등 4개의 서버를 구축한다.

시그널링 서버는 노드 간에 P2P 연결에 필요한 메타데이터와 후보 IP 주소들을 상호 노드들에 전달하기 위한 중계 서버이다.

블록체인 네트워크 관리 서버는 어떤 노드들이 블록체인 네트워크에 참여하고 있는지, 어떤 노드가 네트워크에 참여하고 나갈지를 관리하는 HTTP 서버이다. 또한, 모든 노드가 네트워크에서 나가거나, 다운될 시 모든 데이터가 소실되는 것을 방지하기 위해서 블록체인 사본을 유지한다.

웹 애플리케이션 서버는 구현한 임의의 블록체인 시스템을 시각화 및 WebRTC API가 포함된 프론트엔드와 이 논문에서 제안한 웹 어셈블리 기반의 채굴 모듈을 제공하는 서버이다.

데이터 서버는 임의의 블록에 추가될 데이터를 생성하기 위한 서버로서 해당 서버에서 새로운 데이터가 생성될 때 블록체인 네트워크에 참여하고 있는 모든 노드에게 전파하게 된다.

### 3.4 웹 기반 블록체인 P2P 연결 과정

웹 기반 블록체인 네트워크에 참여하기 위한 P2P 연결과정은 그림 3과 같다. 사용자는 웹 애플리케이션 서버로부터 프론트엔드 페이지를 받으면 하나의 새로운 노드로서 블록체인 네트워크 관리 서버에 웹 기반 블록체인 네트워크에 참여를 요청한다. 블록체인 네트워크 서버는 기존의 블록체인 네트워크에 참여하고 있는 노드들의 UID(Unique Identifier) 정보를 성공 여부와 함께 참여를 요청한 새로운 노드에게 반환한다.

새로운 노드는 WebRTC API를 통해서 자신의 IP 주소, 포트 등이 담긴 SDP(Session Description Protocol) 형식의 메타데이터를 생성한다. 이후 시그널링 서버를 통해서 연결하고자 하는 노드들과 Offer/Answer 메시지를 통해 SDP 메타데이터를 교환한다. 이 과정을 블록체인 네트워크에 참여하고 있는 기존 노드의 수만큼 반복하여 진행한다.

이후 성공적으로 수행되면 각 노드 간에 WebRTC DataChannel이 생성되며 기존의 블록체인 네트워크에 참여 중인 노드와 P2P 연결을 수립하게 된다.

### 3.5 웹 기반 블록체인 네트워크 동작 과정

본 절에서는 웹 기반의 블록체인 네트워크 유지를 위한 각 노드의 동작 과정을 기술한다. 새로운 노드가 웹 기반 블록체인 네트워크에 참여 이후 데이터 동기화 및 작업 증명을 통한 새로운 블록의 검증 과정은 그림 4와 같다.

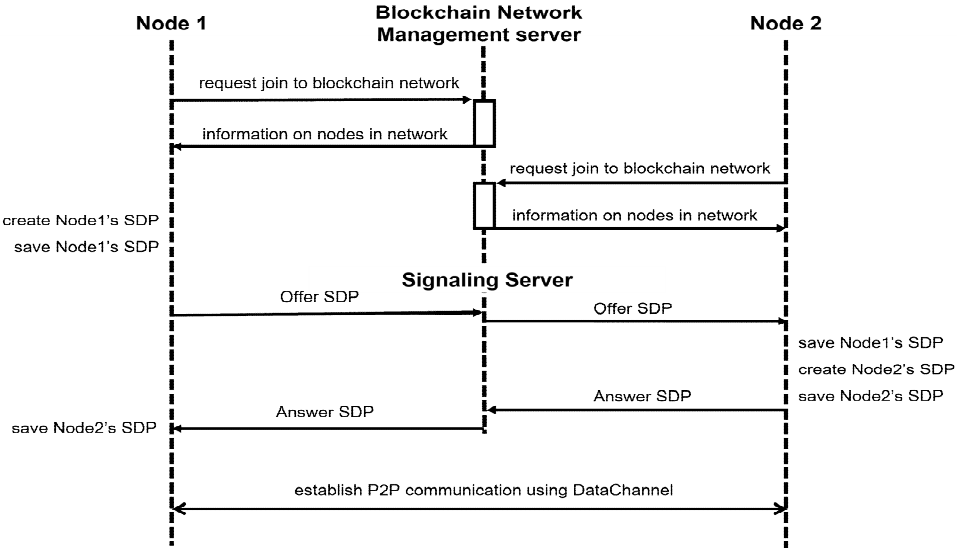


그림 3. 웹 기반의 블록체인 네트워크 참여를 위한 P2P 연결 과정
Fig. 3. P2P connection process for web-based blockchain network participation

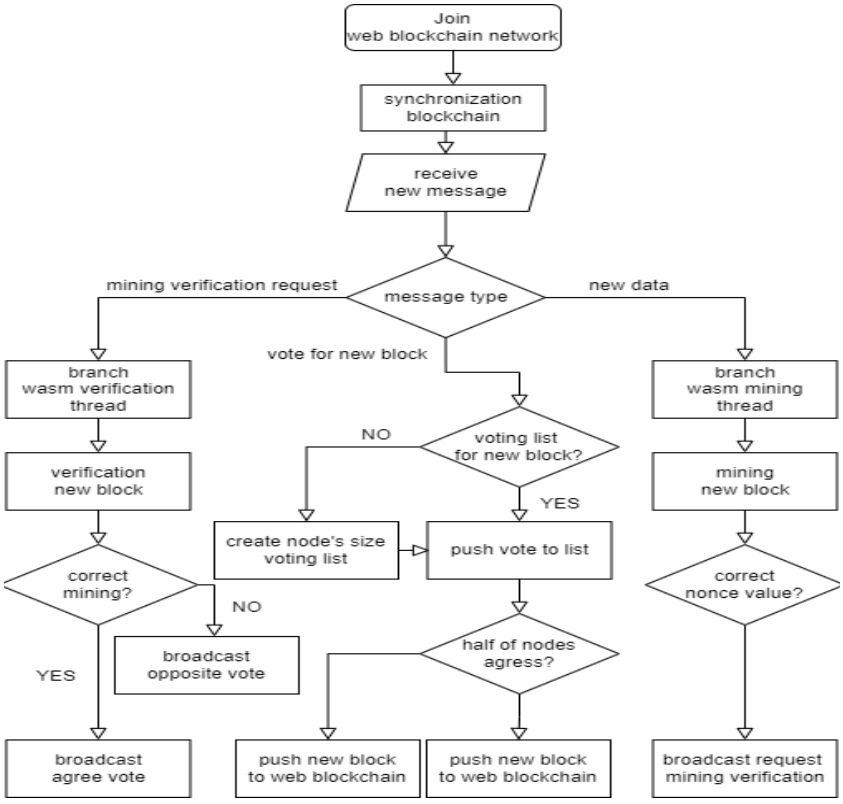


그림 4. 웹 기반의 블록체인 네트워크 동작 및 유지 과정
Fig. 4. Web-based blockchain network operation and maintenance process

새로운 노드는 다른 노드들과 P2P 연결이 확립되면 노드들의 연결 정보가 담긴 리스트를 생성하고 연결된 노드로부터 기존의 블록체인 데이터를 JSON(JavaScript Object Notation) 형태로 전달받아 동기화한다. 이후 새로운 블록 데이터가 발생하기 전까지 대기한다. 이후 블록에 담길 새로운 데이터가 데이터 서버에 생성되면 각 노드는 데이터 서버로부터 양방향 통신이 가능한 웹 소켓(Web socket)을 통해 새로운 블록에 담길 데이터를 전달받으면 각 노드는 웹 워커(Web worker)를 통해서 독립적인

작업 증명 스레드를 분기한다. 작업 증명 스레드에서는 웹 어셈블리 모듈 기반의 해시 연산을 통해 작업 증명을 수행하게 된다.

이후 특정 노드에서 정상적으로 작업 증명을 종료하게 되면 노드는 자신의 정보, 정답 nonce 값, 해당 블록의 해시값, 블록 데이터 정보 등을 블록체인 네트워크 관리 서버와 블록체인 네트워크에 참여 중인 노드들에게 전달한다. 나머지 노드들은 전달받은 채굴 정보를 검증하기 위해서 채굴 검증 스레드를 분기하여 해당 채굴에 대해서 검증을 수행한다.

채굴 검증에 대해서 참으로 판정되면 각 노드는 분기되어 채굴 중인 작업 증명 스레드를 종료하게 된다. 이후 네트워크에 참여한 노드와 블록체인 네트워크 관리 서버에게 해당 작업 증명 정보가 참임을 전달하고 각 노드와 블록체인 네트워크 관리 서버는 네트워크에 참여 중인 노드의 과반수가 해당 정보를 참으로 검증하면 해당 정보를 바탕으로 새로운 블록 생성하여 블록체인에 추가하게 된다. 그림 5는 제안 시스템의 전체 구조도를 보여준다.

## Ⅳ. 실험 및 제안 시스템 구현

본 장에서는 임의의 블록체인의 블록 생성 시간 측정을 통해 웹 어셈블리와 자바스크립트 환경의 성능 차이를 분석하며 제안한 웹 기반의 블록체인 네트워크를 구현 및 시각화 결과를 기술한다.

### 4.1 실험 방법

이 논문에서 구현한 임의의 블록체인 시스템의 난이도 조절은 목표치로 하는 비트의 앞자리 0의 개수에 따라 조절하며 비트의 앞자리를 두 자리, 네 자리, 여섯 자리로 나누어 실험을 수행한다.

성능 측정 실험은 블록체인 채굴 난이도에 따라 블록 생성 시간을 측정한다. 목표로 하는 비트가 두 자리의 실험은 블록의 개수를 1,000개부터 10,000개까지 1,000개 단위로 늘려가며 실험을 진행하고 목표로 하는 비트가 네 자리의 실험은 블록의 개수를 50개부터 500개까지 50개 단위로 늘려가며 진행한다. 마지막으로 목표로 하는 비트가 여섯 자리의 실험은 10개부터 100개까지 10개 단위로 블록의 개수를 늘려가며 실험을 진행한다.

또한 보편적인 실험 결과를 얻기 위해서 각 난이도의 블록 개 수별로 실험을 50회 진행하여 평균 시간을 측정한다.

구현 환경은 표 1과 같다. 웹 환경의 실험을 진행하기 위해서는 Firefox 브라우저 v105.0.1을 사용하였다.

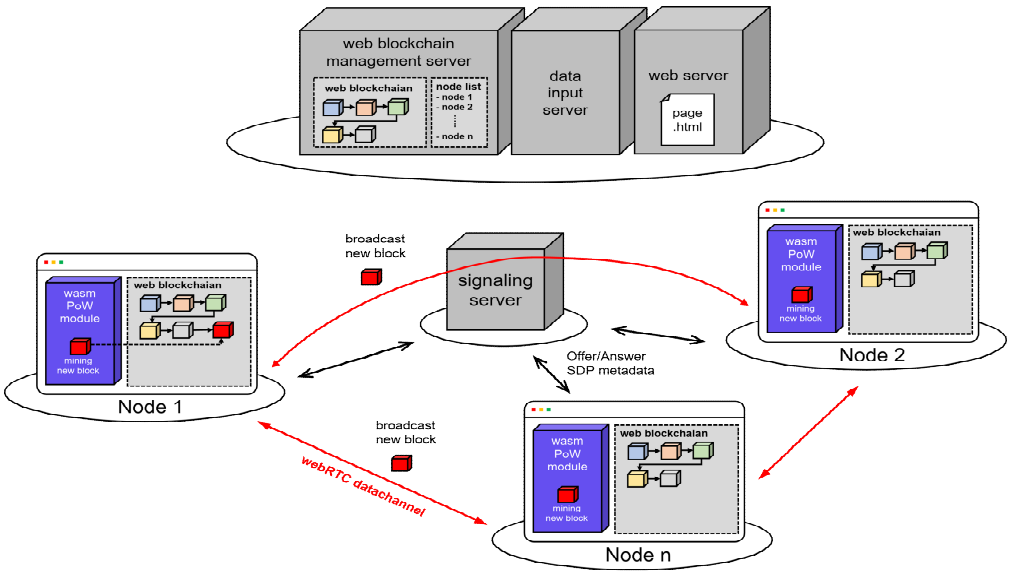


그림 5. 웹 기반의 블록체인 네트워크 시스템 구조도
Fig. 5. Web-based blockchain network system architecture

표 1. 실험 환경
Table 1. Environments for Implementation

| Feature | Specification |
|---|---|
| CPU | Intel(R) Core(TM) i7-7700 @ 3.60GHz |
| OS | Windows 10 Pro |
| Memory | 32GB |
| gcc | 6.3.0 (MinGW.org GCC-6.3.0-1) |
| Emscripten | 2.0.31 |
| Browser | Firefox 105.0.1 (64bit) |

## 4.2 실험 결과

각 환경에서 첫 번째 블록 생성 이후 최종 블록 생성까지의 시간을 측정하기 위해 마지막 블록이 생성된 이후까지 걸린 시간을 출력하여 블록 생성 시간을 측정하였다.

각 난이도 및 블록 개 수별 환경에 따른 최종 블록 생성 시간 통계치는 표 2-4와 같으며 그래프는 그림 6-8과 같다.

첫 번째 실험인 목표로 하는 비트가 두 자리 실험에서는 네이티브 대비 웹 어셈블리 환경이 37.06%, 자바스크립트가 80.36%로 성능 차이를 보였다. 두 번째 실험인 목표로 하는 비트가 네 자리 실험에서는 네이티브 대비 웹 어셈블리 환경이 38.17%, 자바스크립트가 103.45%로 성능 차이를 보였다. 마지막 실험인 목표로 하는 비트가 여섯 자리 실험에서는 웹 어셈블리가 22.22%, 자바스크립트가 86.51%로 성능 차이를 보였다.

표 2. 난이도가 두 자리일 경우 블록 생성 시간
Table 2. Generation time of block with a difficulty of 2 digits

| Blocks \ Env | Native (sec) | WASM (sec) | Javascript (sec) |
|---|---|---|---|
| 50 | 23.607 | 30.224 | 38.793 |
| 100 | 38.977 | 58.198 | 85.903 |
| 150 | 49.749 | 77.429 | 125.648 |
| 200 | 69.325 | 109.085 | 155.823 |
| 250 | 93.585 | 120.595 | 196.35 |
| 300 | 109.435 | 167.992 | 224.752 |
| 350 | 126.977 | 186.338 | 269.676 |
| 400 | 158.387 | 188.91 | 290.543 |
| 450 | 177.086 | 221.044 | 326.187 |
| 500 | 208.219 | 246.022 | 369.121 |

표 3. 난이도가 네 자리일 경우 블록 생성 시간
Table 3. Generation time of block with a difficulty of 4 digits

| Blocks \ Env | Native (sec) | WASM (sec) | Javascript (sec) |
|---|---|---|---|
| 1,000 | 1.178 | 1.863 | 2.114 |
| 2,000 | 1.634 | 3.555 | 5.738 |
| 3,000 | 4.868 | 5.387 | 6.366 |
| 4,000 | 6.243 | 8.435 | 10.627 |
| 5,000 | 8.119 | 9.904 | 12.124 |
| 6,000 | 9.987 | 12.917 | 15.139 |
| 7,000 | 11.036 | 14.141 | 18.04 |
| 8,000 | 12.871 | 15.141 | 20.035 |
| 9,000 | 13.236 | 16.554 | 23.011 |
| 10,000 | 14.266 | 18.112 | 25.421 |

표 4. 난이도가 여섯 자리일 경우 블록 생성 시간
Table 4. Generation time of block with a difficulty of 6 digits

| Blocks \ Env | Native (sec) | WASM (sec) | Javascript (sec) |
|---|---|---|---|
| 10 | 843.307 | 1452.406 | 1795.081 |
| 20 | 1714.42 | 1800.516 | 3519.472 |
| 30 | 2866.49 | 3269.191 | 5287.078 |
| 40 | 3988.44 | 4382.117 | 6488.901 |
| 50 | 6016.05 | 6916.963 | 9296.292 |
| 60 | 7328.22 | 7848.977 | 13372.972 |
| 70 | 7950.18 | 8420.111 | 14483.306 |
| 80 | 8028.75 | 9677.552 | 15894.959 |
| 90 | 8460.3 | 12107.84 | 16927.742 |
| 100 | 11360.1 | 14708.12 | 20743.948 |

모든 난이도의 경우에서 네이티브 환경 대비 웹 어셈블리 환경이 자바스크립트 환경보다 우수한 성능을 보였다. 이는 웹 환경에서 블록체인 시스템을 구축한다면 작업 증명 부분에서 웹 어셈블리가 자바스크립트 대비 높은 효율을 보일 수 있음을 나타낸다.

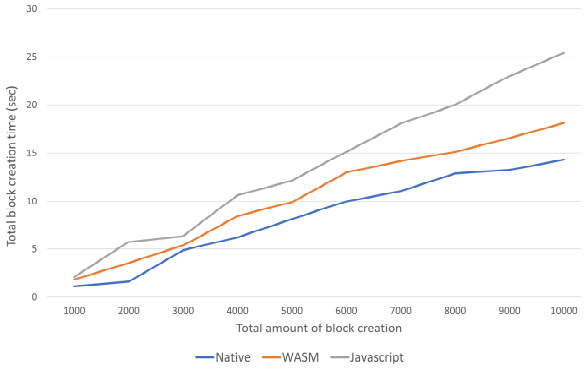


그림 6. 난이도가 두 자리일 때 블록 생성 속도
Fig. 6. Generation time of block with a difficulty of 2 digits

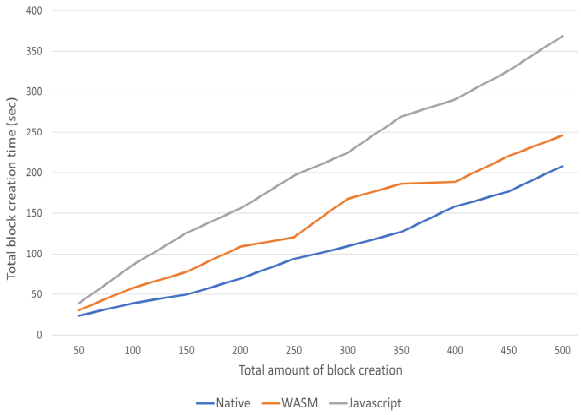


그림 7. 난이도가 네 자리일 때 블록 생성 속도
Fig. 7. Generation time of block with a difficulty of 4 digits

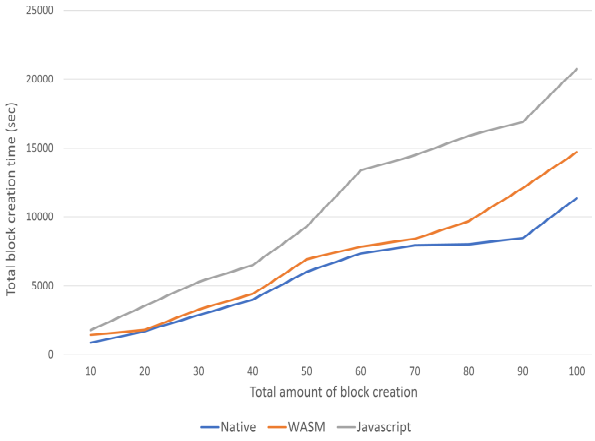


그림 8. 난이도가 여섯 자리일 때 블록 생성 속도
Fig. 8. Generation time of block with a difficulty of 6 digits

## 4.3 웹 기반 블록체인 네트워크 구현 및 시각화

본 절에서는 이 논문에서 제안한 웹 기반 블록체인 네트워크 구현과 구현한 시스템 시각화를 통해 웹에서의 블록체인 활용 가능성을 보고자 한다.

시그널링 서버와 데이터 서버는 node.js와 socket.io 라이브러리를 이용하여 구현한다. 블록체인 네트워크 관리 서버와 웹 애플리케이션 서버는 node.js와 express 프레임워크를 통해 구현한다.

P2P 통신에서 노드가 Private IP를 사용하는 환경과 방화벽 정책이나 NAT(Network Address Translation) 방식에 따라서 시그널링 서버만으로 통신이 제한되는 상황이 생길 수 있다. 이를 대비하기 위해서는 노드의 Public IP와 Port를 알려주는 STUN 서버와 데이터를 중계해주는 TURN 서버를 구축하여야 한다[22]. 본 논문의 실험에서는 Public IP를 발급받은 노드에서만 실험을 진행하므로 STUN과 TURN 서버의 구현은 생략하였다.

그림 9는 세 대의 노드가 참여한 웹 기반 블록체인 네트워크 시각화 화면이다. 세 대의 노드가 웹 기반 블록체인 네트워크 참여하고 Genesis block 생성 이후 데이터 서버로부터 2개의 새로운 데이터를 받아 블록을 채굴 및 검증하고 블록체인 네트워크를 유지하는 것을 확인할 수 있다.

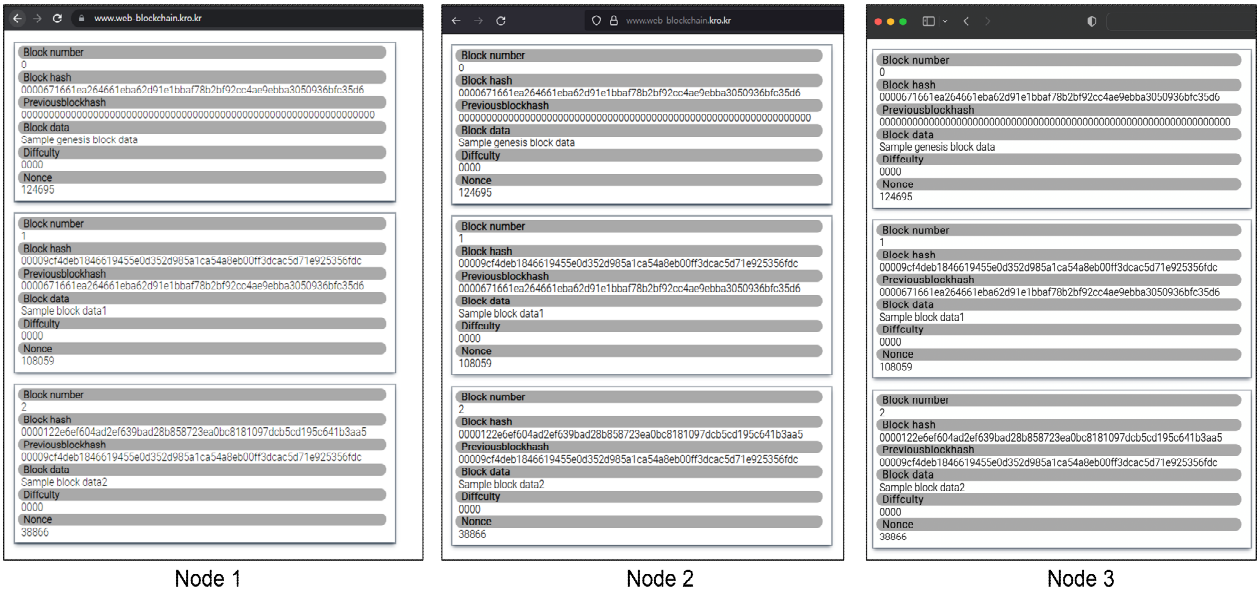


그림 9. 세 대의 노드가 참여한 웹 기반 블록체인 네트워크 시각화 결과
Fig. 9. Web-based blockchain network visualization results

이를 통해 웹 브라우저를 사용하여 사용자에게 블록체인 작업 증명 방법을 제공할 수 있으며 구현한 네트워크를 통해 블록 생성 및 검증하고 블록을 블록체인에 추가할 수 있는 것을 확인할 수 있다.

## Ⅴ. 결론 및 향후 과제

이 논문에서는 웹 환경에서 효율적인 블록체인 작업 증명을 위해 웹 어셈블리와 자바스크립트 환경에서 블록 생성 시간을 측정 및 평가하였다.

환경별 블록 생성 시간 측정 실험 결과 네이티브 대비 웹 어셈블리는 평균 32.48%의 성능 차이를 보였으며 자바스크립트는 평균 90.1%의 성능 차이를 보였다. 이를 통해 웹 환경에서 블록체인 작업 증명에 웹 어셈블리가 자바스크립트 대비 우수한 성능을 보였다.

또한 이 논문에서는 웹 어셈블리 작업 증명 모듈을 활용한 웹 기반 블록체인 네트워크 시스템을 제안하였다. 웹 브라우저 간의 P2P 네트워크를 구현하여 웹 환경에서 블록체인 네트워크를 구축하고 웹 어셈블리 기반의 작업 증명 모듈을 통해 블록 생성 및 검증하는 시스템을 구현하였다.

사용자는 웹 브라우저를 통해 하나의 노드로서 블록체인 네트워크에 참여할 수 있으며 브라우저를 통해 블록을 생성 및 검증과 블록체인 네트워크에 블록을 추가하는 방법을 제공할 수 있다.

추후 웹 기반의 블록체인 네트워크 시스템을 활용한다면 다양한 플랫폼에서 더욱 쉽게 블록체인 네트워크에 참여할 수 있고 이는 블록체인 기술의 편의성 증대와 대중화로 이어질 수 있다. 이는 웹 3.0에 더욱 기반이 되는 하나의 기술로 활용될 수 있을 것으로 기대한다.

향후 연구에서는 네이티브 기반의 블록체인과 웹 기반의 블록체인과의 상호연동 및 활용 방안을 연구하며 웹 환경에서의 작업 증명 효율 향상을 위해 웹 어셈블리의 병렬 계산을 해주는 SIMD(Single Instruction Multiple Data)와 웹 브라우저에서 그래픽 카드 가속 기능을 사용할 수 있는 WebGL(Web Graphics Library)을 활용한 연구를 진행할 예정이다.

## Acknowledgement

이 논문은 2021년도 한국정보기술학회 추계 종합학술대회에서 발표한 논문(웹 어셈블리를 활용한 웹 환경의 블록체인 활용 가능성 분석)을 확장한 논문임[23].

## References

[1] W. S. Lee, M. G. In, S. P. Shin, and K. C. Lee, "Web 3.0 Technology Trends", The Journal of The Korean Institute of Communication Sciences, Vol. 39, No. 6, pp. 42-48, May 2021.

[2] Satoshi Nakamoto, "Bitcoin: a peer-to-peer electronic cash system", Mar. 2009.

[3] S. Tandel and A. Jamadar, "Impact of Progressive Web Apps on Web App Development", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 7, No. 9, pp. 9439-9444, Sep. 2018. https://doi.org/10.15680/IJIRSET.2018.0709021.

[4] J. T. Park and L. Y. Moon, "Development trend of web convergence service implementation technology", The Journal of The Korean Institute of Communication Sciences, Vol. 38, No. 4, pp. 3-9, Mar. 2021. https://doi.org/10.22648/ETRI.2010.J.250601.

[5] A. Zakai, "Emscripten: an LLVM-to-JavaScript compiler", OOPSLA '11: Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion, pp. 301-312, Oct. 2011. https://doi.org/10.1145/2048147.2048224.

[6] J. S. Park, "The Trend of JavaScript in 2020 and beyond – WebAssembly", https://d2.naver.com/helloworld/8257914 [accessed: Oct. 13, 2022]

[7] World Wide Web Consortium, "World Wide Web Consortium(W3C) brings a new language to the Web as WebAssembly becomes a W3C Recommendation", https://www.w3.org/2019/12/

pressrelease-wasm-rec.html.en [accessed: Oct. 02, 2022]

[8] C. H. Shin, J. H. Yeo, and S. M. Moon, "Analysis of Process and Performance in WebAssembly", Communications of the Korean Institute of Information Scientists and Engineers, Jeju, Korea, pp. 1546-1548, Jun. 2018.

[9] "COVID-19 Electronic Vaccination Proof - Coov", https://ncv.kdca.go.kr/coov [Accessed: Oct. 02, 2022]

[10] J. R. Park and S. S. Choi, "Web 3.0 Reboot: Issues and Prospects", Electronics and Telecommunications Trends, Vol. 37, No. 77, pp. 1406-1418, Oct. 2019. https://doi.org/10.22648/ETRI.2022.J.370208.

[11] C. Watt, J. Renner, N. Popescu, S. Cauligi, and D. Stefan, "CT-Wasm: Type-Driven Secure Cryptography for the Web Ecosystem", Proceedings of the ACM on Programming Languages, Vol. 3, No. 77, pp. 1-29, Jan. 2019. https://doi.org/10.1145/3290390.

[12] Y. Oren, V. P. Kemerlis, S. Sethumadhavan, and A. D. Keromytis, "The Spy in the Sandbox: Practical Cache Attacks in JavaScript and their Implications", In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1406-1418, Oct. 2015. https://doi.org/10.1145/2810103.2813708.

[13] J. M. Byeon, "Improvement of Maintenance Equipment Security by Using BlockChain", Journal of the Korea Academia-Industrial cooperation Society, Vol. 22, No. 7, pp. 306-312, Jul. 2021. https://doi.org/10.5762/KAIS.2021.22.7.306.

[14] J. Rüth, T. Zimmermann, K. Wolsing, and O. Hohlfeld, "Digging into Browser-based Crypto Mining", Proceedings of the Internet Measurement Conference, pp. 70-76, Oct. 2018. https://doi.org/10.1145/3278532.3278539.

[15] P. S. Kim, "Analysis standard technology trend of Web service related IETF", The Journal of The Korean Institute of Communication Sciences, Vol. 38, No. 7, pp. 25-32, Jun. 2021.

[16] H. Y. Oh, S. H. Ahn, J. H. Yang, and J. K. Choi, "WebRTC-Based Remote Collaborative Learning Platform", The Journal of Korean Institute of Communications and Information Sciences, Vol. 40, No. 5, pp. 914-923, May 2015. https://doi.org/10.7840/kics.2015.40.5.914.

[17] S. J. Baek, R. H. Lee, and C. S. Yi, "Design and Development of A Systemic Structure to Ensure the Interoperability between the WebRTC-based Video Conferencing Systems and Heterogeneous Terminals", KIISE Transactions on Computing Practices, Vol. 23, No. 4, pp. 238-243, Apr. 2017. https://doi.org/10.5626/KTCP.2017.23.4.238.

[18] A. Chodorek, R. R. Chodorek, and P. Sitek, "UAV-Based and WebRTC-Based Open Universal Framework to Monitor Urban and Industrial Areas", Sensors, Vol. 21, No. 12, pp. 4061, Jun. 2021. https://doi.org/10.3390/s21124061.

[19] C. H. Park and Y. S. Lee, "An Overview of Blockchain Technology: Concepts, Consensus, Standardization, and Security Threats", Journal of the Institute of Convergence Signal Processing, Vol. 20, No. 4, pp. 218-225, 2019. https://doi.org/10.23087/jkicsp.2019.20.4.007.

[20] J. W. Lee and T. Y. Kwon, "Machine Learning Based Prediction of Bitcoin Mining Difficulty", Journal of the Korea Institute of Information Security & Cryptology, Vol. 29, No. 1, pp. 225-234, Feb. 2019. https://doi.org/10.13089/JKIISC.2019.29.1.225.

[21] K. H. Oh and M. Y. Kim, "Blockchain security standardization trend", The Journal of The Korean Institute of Communication Sciences, Vol. 38, No. 4, pp. 45-50, Mar. 2021.

[22] J. H. Woo, I. B. Moon, H. W. Kang, and D. H. An, "Multilingual real-time chat translation and multi-currency application technology based on WebRTC", Communications of the Korean Institute of Information Scientists and Engineers, Jeju, Korea, pp. 1713-1715, Jun. 2022.

[23] W. S. Song, S. W. Jeong, H. J. Jung, and D. W. Jeong, "Analysis of Applicability in Web Assembly-Based Blockchain", Proceedings of KIIT Conference, Jeju, Korea, pp. 255-259, Jun. 2022.

## 저자소개

송 우 석 (Wooseok Song)

2017년 3월 ~ 현재 : 군산대학교 소프트웨어학과 학사과정
관심분야 : 웹, 서버, 블록체인, 딥러닝

정 현 준 (Hyunjun Jung)

2008년 : 삼육대학교 컴퓨터과학과 (학사)
2010년 : 숭실대학교 컴퓨터학과 (공학석사)
2017년 : 고려대학교 컴퓨터·전파 통신공학과(공학박사)
2017년 8월 ~ 2020년 8월 : 광주과학기술원 블록체인인터넷경제연구센터 연구원
2021년 ~ 현재 : 군산대학교 소프트웨어학과 교수
관심분야 : 블록체인, 데이터 사이언스, 센서 네트워크, 사물인터넷, 머신러닝

정 동 원 (Dongwon Jeong)

1997년 2월 : 군산대학교 컴퓨터과학과(학사)
1999년 2월 : 충북대학교 전자계산학과(석사)
2004년 2월 : 고려대학교 컴퓨터학과(박사)
2005년 4월 ~ 현재 : 군산대학교 소프트웨어학과 교수
관심분야 : 데이터베이스, 시맨틱 서비스, 빅데이터, 사물인터넷, 엣지컴퓨팅, 지능형 융합 서비스