ASIC 저항성을 위한 ECCPoW 블록체인 구현 방법

정현준*[1], 채종홍*[2], 이흥노**

Blockchain Implementation Method of Error-Correction Code based Proof-of-Work for ASIC Resistance

Hyunjun Jung*[1], Jong-Hong Chae*[2], and Heung-No Lee**

This work was supported in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-00958) and the National Research Foundation of Korea (NRF) Grant funded by the Korean government (MSIP) (NRF-2018R1A2A1A19018665)

요 약

비트코인은 네트워크에 참여해 금융기관과 같은 제 3자 개입없이 온라인 송금하고 채굴을 하여 보상을 받는다. 비트코인 채굴은 작업증명(Proof-of-Work)을 통해 이뤄지며 작업증명 특성상, 높은 해시 레이트를 가질수록 채굴 확률이 높아진다. 그래서 채굴 조직이라고 불리는 채굴 풀의 등장, CPU/GPU와는 달리 비용과 성능 효율성이 좋은 ASIC 채굴기 등장을 일으켰다. 문제는 이렇게 얻은 해시 레이트로 인해 비트코인의 채굴 독점 문제와 이중 지불 공격 위험에 노출된다. 우리는 ASIC 채굴기의 등장을 해결하기 위해 LDPC 디코더와 해시 함수를 결합한 오류-정정 부호 기반의 작업증명(Error-Correction Codes Proof-of-Work, ECCPoW)을 제안하였다. 이 논문은 ECCPoW의 구현방법에 대하여 제안하고 비트코인의 작업증명을 ECCPoW로 교체했다. 마지막으로 제안방법을 비트코인과 분산성, 보안성, 확장성 측면에서 비교 평가한다.

Abstract

Bitcoin is the first cryptocurrency to participate in a network and receive compensation for online remittance and mining without any third-party intervention, such as financial institutions. Bitcoin mining is done through Proof-of-Work(PoW) and because of its characteristics, the higher hash rate, the higher the probability of mining. Thus, the emergence of a mining pool, which is called a mining organization, and unlike CPU/GPU, ASIC miners with high cost and performance efficiency have emerged. The problem is that the hash rate obtains thus exposes Bitcoin's mining monopoly and the risk of double-payment attack. To solve this problem, we propose Error-Correction Codes Proof-of-Work (ECCPoW) combining the LDPC decoder and hash function. This paper proposes the implementation method of ECCPoW and replaces PoW of bitcoin with ECCPoW. Finally, We compare the proposed method and Bitcoin with decentralization, security, and scalability.

Keywords
proof-of-work, error-correction codes proof-of-work, ECCPoW, ASIC resistance

* 광주과학기술원 블록체인인터넷경제연구센터 연구원
- ORCID[1]: https://orcid.org/0000-0002-6717-1395
- ORCID[2]: https://orcid.org/0000-0003-4235-0271
** 광주과학기술원 블록체인인터넷경제연구센터 (교신저자)
- ORCID : https://orcid.org/0000-0001-8528-5778

· Received: Feb. 20, 2020, Revised: Apr. 22, 2020, Accepted: Apr. 25, 2020
· Corresponding Author: Heung-No Lee
Department of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology, Gwangju 61005, South Korea,
Tel.: +82-62-715-2237, Email: heungno@gist.ac.kr

Ⅰ. 서 론

인터넷 상거래에서 거래할 때 우리는 신뢰 증명을 위하여 제3의 신뢰 기관에 의존한 전자서명을 사용한다. 인터넷상에서 데이터 송수신할 때 위변조에 대한 증명을 중재자에게 위임하여 신뢰를 보증받았다. Satoshi Nakamoto는 비트코인 백서를 통하여 P2P(peer-to-peer network) 네트워크에서 중재자 없는 전자 화폐 시스템을 제안하였다[1]. 비트코인은 블록체인 기술을 전자 화폐 시스템에 적용하여 중재자(예: 은행) 없이 거래의 신뢰를 보증한다. 블록체인은 거래내용이 저장된 블록이 체인 형태의 연결고리 기반 분산 데이터 저장환경에 저장하여 누구라도 임의로 수정할 수 없는 분산 컴퓨팅 기술 기반의 원장 관리 기술이다[2][3].

블록체인은 거래내용이 기록된 원장을 전 세계 네트워크에 분산 저장한다. 블록체인을 유지하기 위하여 블록을 생성한 사람에게 일정한 보상을 지급하도록 설계되었다. 이를 채굴 또는 마이닝(Mining)이라고 하며, 암호화폐(Cryptocurrency)의 거래 내역을 기억한 블록을 생성하고 그 대가로 암호화폐를 얻는 행위를 말한다. 블록체인의 채굴자는 암호화폐의 채굴에 필요한 계산력 향상과 편의성 때문에 채굴 풀(Mining pool)에 소속하여 채굴한다[4][5].

해시 레이트(Hash rate)는 암호화폐를 채굴하기 위한 연산 처리 능력을 측정하는 단위로 초당 해시 값 계산 횟수를 의미한다. 해시 레이트가 높아져 연산량이 많아질 경우 더 빠른 채굴이 가능해지며 블록체인 채굴 난이도는 높아진다. 블록체인에서 채굴 풀이 암호화폐의 전체 해시 레이트의 높은 비율을 차지한다면 이중 지불 공격에 대한 위험이 있다[6]. 소수의 채굴 풀이 전체 해시 레이트의 51%를 점유한다면 블록체인의 분기를 자신이 원하는 쪽으로 결정할 수 있다. 최근 연구에서는 해시 레이트 점유율이 높지 않은 상태에서도 자신의 이득을 위하여 이중 지불 공격을 할 수 있다는 연구결과가 발표되었다[7].

비트코인 채굴자들은 블록체인 정보(version, previous block hash, merkle root, bits 등)를 받아 해시함수(SHA256)의 입력값으로 사용한다. 해시의 출력으로 얻은 값이 현재의 난이도 목푯값보다 작은 경우 블록을 생성할 권리를 갖고 암호화폐를 받는다. 현재 비트코인은 블록을 생성하기 위해 점점 많은 연산이 요구되고 있다. 채굴자들은 비트코인 채굴에 성공하려면 높은 연산을 할 수 있는 ASIC을 구매하여 채굴 풀에 가입한다. 연산량이 낮은 CPU, GPU를 사용하는 채굴자들은 채굴할 기회가 사실상 제로에 가까워졌다.

비트코인은 블록을 생성하기 위하여 SHA256함수에서 나온 출력의 앞자리의 0의 개수를 사용한다. 난이도는 SHA 함수의 아웃풋의 앞자리 0의 개수가 많을수록 높아지고 0의 개수가 작을수록 낮아진다. 채굴자들은 SHA 함수를 더 빠르게 연산하기 위해 ASIC 채굴기를 구매한다. 대중적으로 많이 사용하는 비트메인의 채굴기 Antminer S9(13,000,000 MH/s)는 GPU인 GTX1060(1478 MH/s)과 비교했을 때 약 8,800배 차이가 난다. 최근 발매한 S19은 95 TH/s, S19pro는 110TH/s 성능을 보인다[8]. CPU/GPU를 사용하는 채굴자와 ASIC 칩을 사용하는 채굴자는 채굴에 성공할 확률이 8,800배 차이 난다고 말 할 수 있어서 형평에 어긋난다. 비트코인은 시간이 지날수록 ASIC 칩을 사용한 채굴자에게 화폐가 집중되는 현상이 발생하였다.

블록체인은 노드가 자유롭게 채굴자로 참여하고 공정하게 채굴 보상을 나누고자 제안되었다. 하지만 지금 블록체인은 자유롭게 채굴자로 참여하지 못하고 형평에 어긋나는 경쟁이 이루어지고 있다. ASCI 채굴기의 개발을 억제하기 위한 여러 가지 방법들이 제안되었지만, 결국 ASIC 개발을 막지는 못하였다. 우리는 ASIC 채굴기의 개발을 방지하는 새로운 채굴 함수로써, LDPC(Low Density Parity Check) 디코더와 해시 함수를 결합한 오류-정정 부호 기반의 작업증명(ECCPoW, Error-Correction Codes Proof-of-Work)을 제안하였다[9].

이 논문의 목적은 두 가지이다. 하나는 제안한 ECCPoW를 소개하고 구현방법을 제안한다. 다른 하나는 비트코인에서 SHA256 함수를 ECCPoW 함수로 대체하여 실험한 과정을 소개하는 것이다. 이 논문은 다음과 같이 구성되어 있다. 2장에서는 ASIC 채굴기의 개발을 방지하기 위해 시도된 연구를 소개한다. 3장에서는 ECCPoW를 소개하고 개발방법에 대하여 제시한다. 4장에서는 ECCPoW를 비트코인에

탑재하여 실험한 결과를 소개한다. 5장에서는 구현한 ECCPoW의 성능을 트릴레마 문제인 분산성, 보안성, 확장성 부분에서 평가한다. 마지막으로 6장에서는 결론을 제시한다.

II. 관련 연구

2.1 이더리움

이더리움은 블록체인 기술을 기반으로 스마트 계약 기능을 구현하기 위해 개발된 분산 컴퓨팅 플랫폼이다. 2015년 7월 비탈릭 부테린이 C++과 Go 언어로 개발했다. 이더해시(Ethash)알고리즘 기반의 작업증명 방식으로 채굴 중이지만, 앞으로 작업증명 방식을 지분증명(PoS) 방식으로 변경할 예정이다[10].

이더리움은 비선형 그래프(DAG, Directed Acyclic Graph)를 이용하여 ASIC에 대항한다. DAG의 초기 크기는 약 1GB 이였으며, 천천히 시간이 지날수록 선형으로 크기가 증가하도록 설계되었다. 2019년 10월 현재 DAG의 크기는 3.99GB이며 2020년 12월 20일까지 유지된다[11][12].

이더리움은 2019년 ASIC의 채굴 중앙화에 대응하기 위하여 ProgPoW[13]를 개발하여 적용하기로 승인하였다. ProgPoW는 다음과 같은 특징을 가지고 있다. 첫째, 채굴 시 정기적으로 문제를 변경한다. 두 번째, 채굴에 그래픽 카드의 모든 구성 요소를 최대한 활용한다. ProgPoW는 블록 번호를 기반으로 임의 생성된 문제를 사용하고 GPU의 효율적인 작동을 위해 설계하였다. 그리고 정기적으로 GPU가 빠르게 적응할 수 있는 문제로 변경하면서 ASIC에 대비 성능 차이를 줄인다.

2.2 X-11 계열

X-11은 뒤에 붙은 숫자만큼의 해시 함수를 사용하는 암호화폐 채굴 알고리즘이다[14]. X11은 ASIC을 억제하기 위해 다수의 해시 함수를 사용하여 심층과 복잡성을 추가했다. 대표적으로 대쉬(Dash)에서 사용하고 있다. X11은 여러 해시 함수를 연결하여 해시의 출력값이 다음 해시의 입력값으로 사용한다.

X-11 계열 개념은 여러 개의 해시를 사용하여 보안성은 높이고 ASIC 채굴을 막는 것이다. 하지만 현재, ASIC 채굴이 가능하게 되면서 X13, X14, X15, X16R, X17 등 알고리즘 숫자를 증가하는 업그레이드가 나왔다. 하지만 이에 대응하는 ASIC이 등장하였다.

2.3 크립토노트(CryptoNote)

크립토노트는 ASIC 채굴을 방지하기 위하여 CPU보다 GPU에서의 실행을 비효율적으로 설계하였다[15]. 크립토노트 성능은 메모리 작성과 후속 읽기 작업이 반복적으로 발생하기 때문에 메모리 지연 시간에 매우 민감하다. 이는 DAG를 사용하는 이더리움의 Ethash함수와 유사하다. 메모리 집약적인 작업 결과 이후에 사용할 해시 함수를 결정하여 최종적인 블록을 생성한다.

이러한 시도에도 불구하고, 2018년도 3월 비트메인에서 크립토노트 채굴 알고리즘에 최적화된 채굴기를 발매했다. 크립토노트를 적용한 모네로는 이를 막기 위해 일 년에 두 번씩 채굴 알고리즘을 변경하는 방안을 시행 중이다. 하지만 잦은 하드포크는 참여자들이 네트워크에서 이탈하는 상황을 만들고 채굴의 집중화를 가져오는 위험이 발생하였다. 잦은 하드포크를 방지하기 위하여 RandomX는 주기적으로 채굴 방법을 변경하는 키 블록개념을 제안하였다[16].

III. ECCPoW Blockchain 구현방법

이 장에서는 제안한 ECCPoW의 개요와 구현방법에 대하여 설명한다. 우리는 ASIC 저항성을 높이기 위하여 ECCPoW의 개념을 제안하였다. 해시 함수의 ASIC 저항성이란 ASIC 개발이 어렵게 하는 해시 함수의 성질이다. 기존 ASIC 저항성 연구는 이더리움과 같이 메모리의 적재를 유도하는 방법과 X11과 같이 여러 해시 알고리즘을 사용하는 방법이다. 하지만 이더리움의 DAG 방식과 X-11 계열의 방식은 ASIC 장비가 등장하였다. 해시 알고리즘을 일정 주기로 변경하는 크립토노트와 같은 방법은 잦은 하드포크로 인하여 사용자는 불편함을 느끼고 채굴이

점점 줄어들었다. ECCPoW는 ASIC 등장을 억제하기 위하여 매 블록 새로운 함수를 자동으로 정의하는 방법을 제안한다.

3.1 ECCPoW 개요

ECCPoW는 통신에서 많이 사용되는 오류-정정 부호 (error-correction codes)의 디코더 (decoder)를 활용한 작업증명이다. 오류-정정 부호에서 사용되는 디코더는 일반적으로는 ASIC 장치를 이용해 구현할 수 있다. 간단한 예로써, 우리가 쓰는 핸드폰에도 오류-정정 부호 디코더를 빠르고 저전력으로 구현하기 위해 ASIC를 이용한다. ASIC 기반의 오류-정정 부호 디코더의 설계는 디코더의 입력값 중의 하나인 패리티 체크 행렬 H에 의해 결정된다. 즉 패리티 체크 행렬을 고정하면 디코더를 ASIC 장비로 제작할 수 있다. 핸드폰의 경우, 표준화된 패리티 체크 행렬들이 결정돼있어 이를 위한 ASIC 기반의 오류-정정 부호 디코더 설계가 가능하다. 하지만, 무수히 많은 패리티 체크 행렬들을 지원하는 디코더에 맞춰 ASIC 장치를 제작하는 것은 비용 문제 및 디코더의 크기 문제 등 현실적으로 불가능하다.

ECCPoW 작업증명에서는 매 블록 패리티 체크 행렬이 무작위로 바뀌게 한다. 다시 말해 ECCPoW의 작업증명이 사용하는 패리티 체크 행렬의 개수는 무한하다고 할 수 있다. 이렇게 함으로써, 오류-정정 부호 디코더를 위한 ASIC 장치 개발을 억제한다. 오류-정정 부호에서 실행하는 디코딩 알고리즘을 CPU 혹은 GPU 로만 실행하게 된다. 결국, 기존 작업증명에서 사용되는 SHA 함수를 빠르게 실행하더라도, 오류-정정 부호 디코더의 디코딩 알고리즘 실행에서 병목현상이 발생한다. 즉, ASIC 장치의 사용을 억제할 수 있다.

3.2 매 블록마다 바뀌는 암호 퍼즐 생성

ECCPoW 작업증명에서는 매 블록마다 바뀌는 암호 퍼즐을 생성하고자 한다. Gallager[17]의 생성 방법과 이전 해시값을 동시에 사용해 암호 퍼즐 생성에 사용되는 합성함수를 매 블록 마다 변경하였다. 즉, 합성함수의 부호 디코더가 사용하는 LDPC(Low Density Parity Check) 패리티 체크 행렬 H를 매 블록마다 무작위로 생성되게 구현하였다. Gallager의 방법을 이용하기 위해서는 변수들이 필요하다. 표 1은 LDPC에서 사용하는 변수의 의미를 보여준다.

표 1. LDPC 패리티 체크 행렬의 변수
Table 1. Variables in the LDPC parity check matrix

Variables	Characteristics
n	Number of columns in H
m	Number of rows in H
w_c	H Number of 1 in each column
w_r	H Number of 1 in each row

이 변수들은 $nw_c=(n-k)w_r$를 만족해야 한다. 여기서 k는 $n-m$이고, 2^k는 생성 가능한 부호의 총 개수이다. 변수들이 주어졌을 때, 크기가 $m\times n$인 LDPC 패리티 체크 행렬 H를 다음 방법에 의해 생성한다.

Step 1: 크기가 $\frac{m}{w_c}\times n$인 부분 행렬 생성

$$A_1 := \begin{bmatrix} \underbrace{1\ 1\ \cdots\ 1}_{w_r} & & & \\ & \underbrace{1\ 1\ \cdots\ 1}_{w_r} & & \\ & & \ddots & \\ & & & \underbrace{1\ 1\ \cdots\ 1}_{w_r} \end{bmatrix} \in \{0,1\}^{\frac{m}{w_c}\times n} \quad (1)$$

Step 2: 상기 행렬을 임의로 순열시켜 w_c-1개의 부분 행렬들을 각각 생성

$$A_i := \Pi_i(A_i) \in \{0,1\}^{\frac{m}{w_c}\times n} \quad (2)$$

여기서 Π_i는 i번째 순열시키는 순서이고, $i=2,3,\cdots,w_c$이다.

Step 3: 상기 모든 부분 행렬들을 이용해 최종 LDPC 행렬을 구성

$$H := \begin{bmatrix} A_1^T & A_2^T & \cdots & A_{w_c}^T \end{bmatrix} \in \{0,1\}^{m\times n} \quad (3)$$

ECCPoW는 순열 순서를 이전 해시값을 통해 변

경되게 한다. 이전 해시값을 시드 값으로써 활용하여 순열 순서를 결정한다. 해시값은 무작위의 값이므로 순열 순서는 무작위가 된다. [18]에서 구현한 코드를 확인할 수 있다. 표 2는 서로 다른 이전 해시값을 사용 시, 생성된 H를 비교하였다. 빨간색 쳐진 부분이 서로 다른 것을 확인할 수 있다.

표 2. 서로 다른 해시값을 사용했을 때의 생성된 H의 형태
Table 2. Form of the resulting H of using a different hash value

Generated H $n=24$ $m=16$ $w_c=3$ $w_r=4$	
Previous hash value	0x00000000000000000000000000000001
생성된 H $n=24$ $m=16$ $w_c=3$ $w_r=4$	
Previous hash value	0x00000000000000000000000000000002

3.2 매 블록마다 바뀌는 암호 퍼즐 디코더

ECCPoW의 LDPC 디코더는 메시지 전달 기반 알고리즘(massage-passing algorithm)을 이용하여 개발하였다. 디코더는 길이가 n인 해시값 $r \in \{0,1\}^n$과 $m \times n$인 LDPC 행렬 H를 입력값으로 취득하여 길이가 n인 출력값 $c \in \{0,1\}^n$를 산출한다.

디코더는 입력받은 해시값 r에 따라 2가지 형태의 답을 산출할 수 있다.

입력된 해시값 r이 임의의 부호 c_i에 대해 $\| r - c_i \|_h \leq t$를 만족하면 디코더는 해당 부호 $D_{MP}: \{r, H\} \mapsto c_i$를 산출한다. 여기서 t는 LDPC 행렬 H에 의해 결정되는 값이다. 만약 만족하지 않으면 디코더는 임의의 벡터 $c \in \{0,1\}^n$ 를 산출한다. [18]에서 실제 구현된 디코딩 함수의 소스 코드를 확인할 수 있다.

표 3. ECCPoW 암호 퍼즐 해결 유무 판단 기준
Table 3. Criteria for the determination of ECCPOW crypto puzzle resolution

Condition 1	(Original method) The result of the decoder is the code, and if you have the specific hamming weight is determined to solve the problem.
Condition 2	(Existing Proof of Work) If the result of rehashing the result of the decoder is less than a specific value, it is determined that the problem is solved.

암호 퍼즐 해결 여부를 판단하는 기준은 표 3의 2가지 기준들을 사용한다. 기준 1은 디코더의 출력값 c이 조건들을 만족시키면 암호 퍼즐을 해결한 것으로 판단한다.

조건 1 - 출력값이 부호임.

조건 2 - 출력값의 해밍 가중치가 주어진 집합 S의 원소임

조건 1은 디코더가 임의의 입력값을 받았을 때, 부호를 산출할 확률이 적다는 것에서 기인한다. 조건 2는 H가 주어졌을 때 생성 가능한 부호들의 해밍 가중치들이 다를 수 있다는 것에서 기인한다.

조건 1을 만족할 확률을 구하려면 H의 최소 해밍 거리 값이 필요하다. 이 값을 계산하려면 2^k의 서로 다른 부호들을 모두 고려해야 한다. 부호의 개수가 작을 때에는 가능하지만 개수가 클 때는 불가능하다. Litsyn[19]은 특정 w_c, w_r일 때, H의 최소 해밍 거리 값의 상한/하한값들을 보고하였다.

표 4. LDPC 패리티 체크 행렬의 변수에 따른 부호를 발견할 확률
Table 4. Probability of finding a sign according to the variable of LDPC parity check matrix

$w_c=4,\ w_r=5$	p_1 Upper bounds	p_1 Lower bounds
$n=80,\ k=12$	6.32×10^{-5}	2.12×10^{-8}
$n=120,\ k=24$	1.65×10^{-8}	1.49×10^{-13}
$n=160,\ k=32$	4.06×10^{-10}	1.34×10^{-17}

표 4는 LDPC 패리티 체크 행렬의 변수에 따른 부호를 발견할 확률을 보여준다. 이를 보면, 확률의

상한값이 매우 작은 것이 확인된다. 이는 임의의 값을 취득했을 때 디코더가 조건 1을 만족할 확률이 작다는 것을 뜻한다.

조건 2는 변수들 n, m, w_c, 그리고 w_r이 고정되었을 때, 암호 퍼즐의 난이도를 높이기 위하여 사용된다. 표 5는 $n=256$, $m=192$, $w_c=4$, $w_r=5$일 때 생성 가능한 부호들의 해밍 가중치의 분포도의 일부와 집합 S가 주어졌을 때 조건 2가 만족할 확률값이다.

표 5. 조건 2가 만족할 확률
Table 5. Probability that condition 2 will satisfy

Hamming weight	Probability	Element of the set S	Probability that condition 2 will satisfy
98	$\approx 5\times10^{-5}$	98	$\approx 5\times10^{-5}$
…	…	…	
126	$\approx 9.7\times10^{-2}$	98,000, …, 126	$\approx 4\times10^{-1}$
128	$\approx 1\times10^{-1}$	98,000, …, 126,128	$\approx 5\times10^{-1}$

조건 1과 조건 2 모두를 동시에 만족할 확률은 다음과 같다.

$$p := \Pr\{c|Hc=0\}\times\Pr\{\|c\|_h \in S\} \quad (4)$$

변수들 n, w_c, w_r과 집합 S가 주어졌을 때 조건 1과 조건 2를 동시에 만족할 확률을 계산하여 표 6과 같은 난이도 테이블을 제작하였다. 표 6에 있는 확률값 p는 암호 퍼즐 난이도를 뜻한다. 확률값이 0에 가까울수록 암호 퍼즐 난이도가 높음을 뜻한다. p의 역수값은 암호 퍼즐을 풀기 위한 시도 횟수의 기댓값이다.

기준 2는 디코더의 출력값과 논스를 다시 해싱하여 나온 결괏값을 얻고, 해당 결괏값이 정해진 타겟(Target)과 비교하여 암호 퍼즐의 해결 여부를 판단한다. 그림 1은 ECCPoW 암호 퍼즐 해결 유무 판단 기준 2를 표현한다. 합성 함수와 해시 알고리즘을 하나의 해시 함수로 인식하면 그림 3은 비트코인과 동일한 구조이므로, 비트코인의 난이도 조절 함수를 사용할 수 있다.

표 6. ECCPoW의 난이도 테이블
Table 6. Difficulty table of ECCPoW

Lv.	n	w_c	w_r	Set S	p
1	32	3	4	{10, 12,…, 20, 22}	$\approx 3.07\times10^{-5}$
2	32	3	4	{10, 12,…, 14, 16}	$\approx 2.02\times10^{-5}$
…					
379	128	3	4	{34, 94}	$\approx 5.12\times10^{-23}$
380	128	3	4	{34}	$\approx 2.60\times10^{-23}$

Ⅳ. 실 험

이 장에서는 3장을 통해 설계된 ECCPoW를 검증하기 실험을 하였다. 단일 노드 실험에서는 비트코인 합의 알고리즘을 ECCPoW로 교체하여 블록생성 기능을 확인한다. 복수 노드 실험에서는 복수의 노드 환경에서 블록생성, 블록 동기화 그리고 트랜잭션 생성 및 전송이 제대로 이뤄지는지 확인하는 실험을 진행하였다.

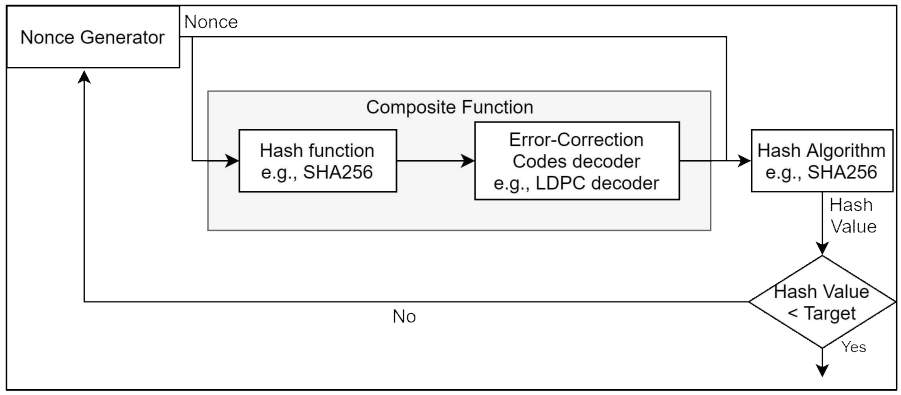
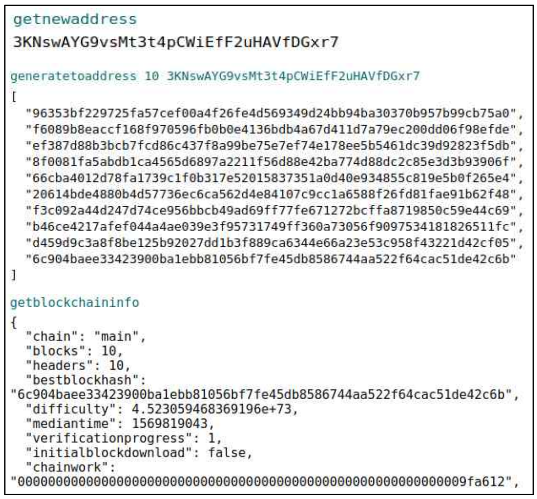


그림 1. ECCPoW 암호 퍼즐 해결 유무 판단 기준 2
Fig. 1. Criteria 2 for ECCPOW crypto puzzle resolution determination

4.1 단일 노드 실험

비트코인의 합의 알고리즘을 ECCPoW로 교체하였다. 단일 노드 실험은 내부망에서 ECCPoW 블록체인이 제대로 동작하는지 블록생성 실험이다.

블록생성을 위해서는 “generatetoaddress” 명령어를 사용하는데 매개변수 값으로 블록체인의 현재 주소가 들어간다. 따라서 “getnewaddress” 명령어를 이용하여 블록체인의 주소를 새로 생성하고 “generatetoaddress”를 이용하여 블록 10개를 생성했다. 생성 후 블록체인의 정보를 확인한 결과 “blocks”의 정보가 10개로 변화됨을 그림 2에서 확인할 수 있다.



```
getnewaddress
3KNswAYG9vsMt3t4pCWiEfF2uHAVfDGxr7

generatetoaddress 10 3KNswAYG9vsMt3t4pCWiEfF2uHAVfDGxr7
[
  "96353bf229725fa57cef00a4f26fe4d569349d24bb94ba30370b957b99cb75a0",
  "f6089b8eaccf168f970596fb0b0e4136bdb4a67d411d7a79ec200dd06f98efde",
  "ef387d88b3bcb7fcd86c437f8a99be75e7ef74e178ee5b5461dc39d92823f5db",
  "8f0081fa5abdb1ca4565d6897a2211f56d88e42ba774d88dc2c85e3d3b93906f",
  "66cba4012d78fa1739c1f0b317e52015837351a0d40e934855c819e5b0f265e4",
  "20614bde4880b4d57736ec6ca562d4e84107c9cc1a6588f26fd81fae91b62f48",
  "f3c092a44d247d74ce956bbcb49ad69ff77fe671272bcffa8719850c59e44c69",
  "b46ce4217afef044a4ae039e3f95731749ff360a73056f9097534181826511fc",
  "d459d9c3a8f8be125b92027dd1b3f889ca6344e66a23e53c958f43221d42cf05",
  "6c904baee33423900ba1ebb81056bf7fe45db8586744aa522f64cac51de42c6b"
]

getblockchaininfo
{
  "chain": "main",
  "blocks": 10,
  "headers": 10,
  "bestblockhash":
"6c904baee33423900ba1ebb81056bf7fe45db8586744aa522f64cac51de42c6b",
  "difficulty": 4.5230594683691 96e+73,
  "mediantime": 1569819043,
  "verificationprogress": 1,
  "initialblockdownload": false,
  "chainwork":
"0000000000000000000000000000000000000000000000000000000000009fa612",
```


그림 2. 블록생성 결과
Fig. 2. Results of block generation

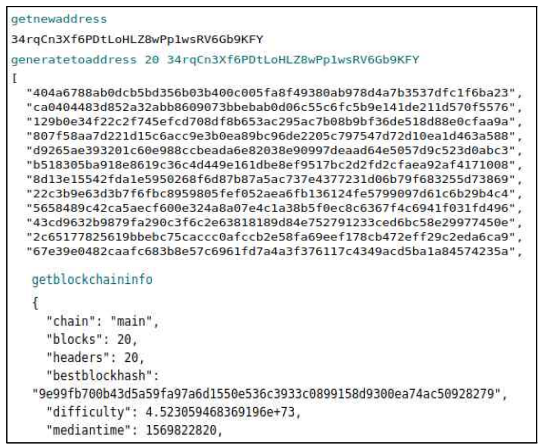
```
getnewaddress
3FUFZ8ShBQpCGqqkZoFErG2aGo28QwaUDT

generatetoaddress 10 3FUFZ8ShBQpCGqqkZoFErG2aGo28QwaUDT
[
  "2b2d7485a399d93169ff8532de059db263e0ba42db73847489d4620a7b7510bf",
  "099640a372ec928e5af6aa1b2032014b24a04225724c5553c1a80bf53dd0d8c4",
  "1c44c7078afcdd49eb9ab8fd4ccbc64419cca96afda91c571a51de163b0ac9ce",
  "630b1b3127c74b8892ff7fcbae89f2733aeadf57f225354a05f60583200ac8a2",
  "74449f452fd35732dbfab5e3ada8f531d7df4ee1a7df23ef41ab91ca1b4c3ca3",
  "282768f094a38a6f7113e4ffc45f5cb93ae7fb01f9e164bd8482f59df49712ff",
  "4a16965fc7efab971b99cbd8cd807eb78b17852fe4be55a620cf00ca865bbafc",
  "5ea006ef89893ab591ba39d995c2b09bcba5ac21813a3c8ff9d7752765c820e7",
  "c43c6bb01c954f23b998f88ead512a7b7cbc1350e37225e6659f02f923ea8f17",
  "7a39ec86ae15a01d8d6930f2fcc5300d8830726fab16b4e733c59db1719f8e43"
]

getblockchaininfo
{
  "chain": "main",
  "blocks": 10,
  "headers": 10,
  "bestblockhash":
"7a39ec86ae15a01d8d6930f2fcc5300d8830726fab16b4e733c59db1719f8e43",
  "difficulty": 4.5230594683691 96e+73,
  "mediantime": 1569822743,
```


그림 3. node 1의 블록 채굴 및 확인
Fig. 3. Block mining and verification in node 1

4.2 복수 노드 실험

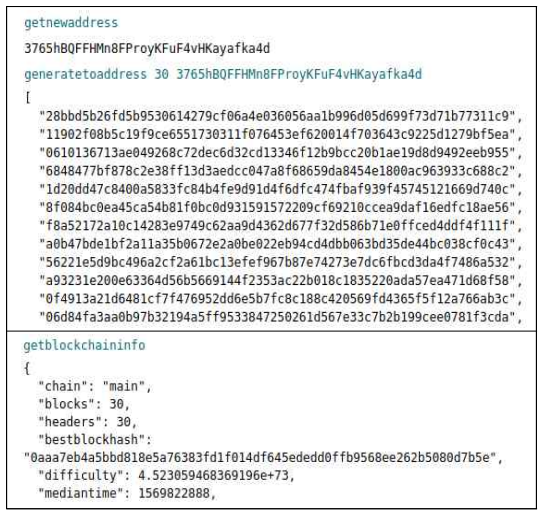
복수 노드 실험은 임의의 노드 3개(node 1, node 2, node 3)를 이용하여 블록 동기화 실험, 트랜잭션 전송 및 확인을 한다.

각 노드에 특정 블록 수를 채굴하여 긴 블록체인에 동기화가 되는지 확인한다. node 1에는 블록 10개, node 2에는 블록 20개, 그리고 node 3에는 블록 30개를 생성한다. 그림 3은 node 1에서 블록을 채굴하고 확인한 결과, 그림 4는 node 2에서 블록을 채굴하고 확인한 결과와 그림 5는 node 3에서 블록을 채굴하고 확인한 결과를 나타내고 있다.



```
getnewaddress
34rqCn3Xf6PDtLoHLZ8wPp1wsRV6Gb9KFY
generatetoaddress 20 34rqCn3Xf6PDtLoHLZ8wPp1wsRV6Gb9KFY
[
  "404a6788ab0dcb5bd356b03b400c005fa8f49380ab978d4a7b3537dfc1f6ba23",
  "ca0404483d852a32abb8609073bbebab0d06c55c6fc5b9e141de211d570f5576",
  "129b0e34f22c2f745efcd708df8b653ac295ac7b08b9bf36de518d88e0cfaa9a",
  "807f58aa7d221d15c6acc9e3b0ea89bc96de2205c797547d72d10ea1d463a588",
  "d9265ae393201c60e988ccbeada6e82038e90997deaad64e5057d9c523d0abc3",
  "b518305ba918e8619c36c4d449e161dbe8ef9517bc2d2fd2cfaea92af4171008",
  "8d13e15542fda1e5950268f6d87b87a5ac737e4377231d06b79f683255d73869",
  "22c3b9e63d3b7f6fbc8959805fef052aea6fb136124fe5799097d61c6b29b4c4",
  "5658489c42ca5aecf600e324a8a07e4c1a38b5f0ec8c6367f4c6941f031fd496",
  "43cd9632b9879fa290c3f6c2e63818189d84e752791233ced6bc58e29977450e",
  "2c65177825619bbebc75caccc0afccb2e58fa69eef178cb472eff29c2eda6ca9",
  "67e39e0482caafc683b8e57c6961fd7a4a3f376117c4349acd5ba1a84574235a",

getblockchaininfo
{
  "chain": "main",
  "blocks": 20,
  "headers": 20,
  "bestblockhash":
"9e99fb700b43d5a59fa97a6d1550e536c3933c0899158d9300ea74ac50928279",
  "difficulty": 4.5230594683691 96e+73,
  "mediantime": 1569822820,
```


그림 4. node 2의 블록 채굴 및 확인
Fig. 4. Block mining and verification in node 2



```
getnewaddress
3765hBQFFHMn8FProyKFuF4vHKayafka4d
generatetoaddress 30 3765hBQFFHMn8FProyKFuF4vHKayafka4d
[
  "28bbd5b26fd5b9530614279cf06a4e036056aa1b996d05d699f73d71b77311c9",
  "11902f08b5c19f9ce6551730311f076453ef620014f703643c9225d1279bf5ea",
  "0610136713ae049268c72dec6d32cd13346f12b9bcc20b1ae19d8d9492eeb955",
  "6848477bf878c2e38ff13d3aedcc047a8f68659da8454e1800ac963933c688c2",
  "1d20dd47c8400a5833fc84b4fe9d91d4f6dfc474fbaf939f45745121669d740c",
  "8f084bc0ea45ca54b81f0bc0d931591572209cf69210ccea9daf16edfc18ae56",
  "f8a52172a10c14283e9749c62aa9d4362d677f32d586b71e0ffced4ddf4f111f",
  "a0b47bde1bf2a11a35b0672e2a0be022eb94cd4dbb063bd35de44bc038cf0c43",
  "56221e5d9bc496a2cf2a61bc13efef967b87e74273e7dc6fbcd3da4f7486a532",
  "a93231e200e63364d56b5669144f2353ac22b018c1835220ada57ea471d68f58",
  "0f4913a21d6481cf7f476952dd6e5b7fc8c188c420569fd4365f5f12a766ab3c",
  "06d84fa3aa0b97b32194a5ff9533847250261d567e33c7b2b199cee0781f3cda",

getblockchaininfo
{
  "chain": "main",
  "blocks": 30,
  "headers": 30,
  "bestblockhash":
"0aaa7eb4a5bbd818e5a76383fd1f014df645ededd0ffb9568ee262b5080d7b5e",
  "difficulty": 4.5230594683691 96e+73,
  "mediantime": 1569822888,
```


그림 5. node 3의 블록 채굴 및 확인
Fig. 5. Block mining and verification in node 3

node 3에서 “addnode"를 이용하여 node 1(현재 IP 주소, 192.168.232.128)과 node 2 (현재 IP주소, 192.168.232.129)를 연결 후 "getaddednodeinfo”를 통해 연결된 노드 정보를 확인한다. 이러한 결과를 그림 6에서 나타내고 있다. 이렇게 연결된 노드들은 많은 블록을 보유하고 있는 블록체인을 우선으로 동기화한다. 블록체인이 많은 블록을 보유하고 있다는 건 블록체인이 크다 할 수 있으며 이는 해당 블록체인에 대한 신뢰성이 많다고 판단하기 때문이다. 현재 시험에서는 30개를 채굴한 node 3이 제일 많은 블록을 보유하고 있어서 node 1과 node 2는 node 3의 블록을 우선으로 동기화할 것이며 이러한 결과를 그림 7에서 보여주고 있다.

```
addnode 192.168.232.128 add
null
addnode 192.168.232.129 add
null
```

```
getaddednodeinfo
[
  {
    "addednode": "192.168.232.128",
    "connected": true,
    "addresses": [
      {
        "address": "192.168.232.128:9777",
        "connected": "outbound"
      }
    ]
  },
  {
    "addednode": "192.168.232.129",
    "connected": true,
    "addresses": [
      {
        "address": "192.168.232.129:9777",
        "connected": "outbound"
      }
    ]
  }
]
```

그림 6. node 3의 노드 연결
Fig. 6. Node connections in node 3

```
getblockchaininfo
{
  "chain": "main",
  "blocks": 30,
  "headers": 30,
  "bestblockhash":
"0aaa7eb4a5bbd818e5a76383fd1f014df645ededd0ffb9568ee262b5080d7b5e",
  "difficulty": 4.523059468369196e+73,
  "mediantime": 1569822888,
```

그림 7. node 1과 node2의 블록체인 동기화
Fig. 7. Blockchain synchronization of node 1 and node 2

동기화 확인으로 블록체인이 제대로 동작을 확인할 수 있었다. 이렇게 연결된 node 2와 node 3간의 트랜잭션을 전송하여 확인하는 실험을 했다. 그림 8은 node 2와 node 3이 보유하고 있는 금액을 나타내고 있다. 현재, node 2는 0을 가지고 있고 node 3은 1000을 가지고 있다.

그림 9는 node 3이 node 2에 500코인을 전송하기 위해 node 2의 주소를 적고(Pay-To), 보낼 코인 양(Amount)을 적었다. 전송 비용(Transaction Fee)은 최소 비용인 0.00001로 설정하였다. 전송이 완료된 다음 그림 10에서는 node 2의 코인 양(Balance)과 최근 트랜잭션 기록(Recent transactions)을 확인하여 트랜잭션이 제대로 전송되었음을 알 수 있다.

```
getbalance
0.00000000
```

```
getbalance
1000.00000000
```

그림 8. node 2(위), node 3(아래)의 금액
Fig. 8. Balance of node 2 (top), node 3 (bottom)

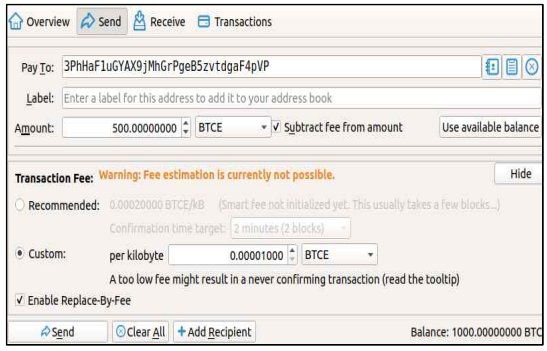


그림 9. 트랜잭션 전송 입력
Fig. 9. Input of transaction transfer

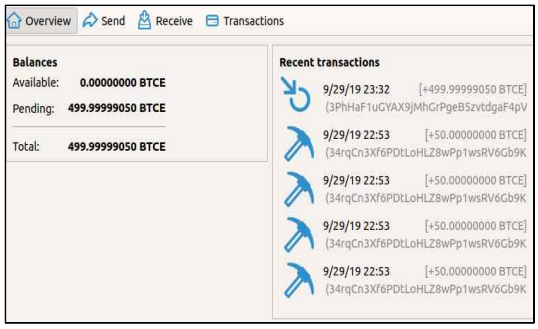


그림 10. node 2의 금액과 트랜잭션 기록
Fig. 10. Balances and transaction logs in node 2

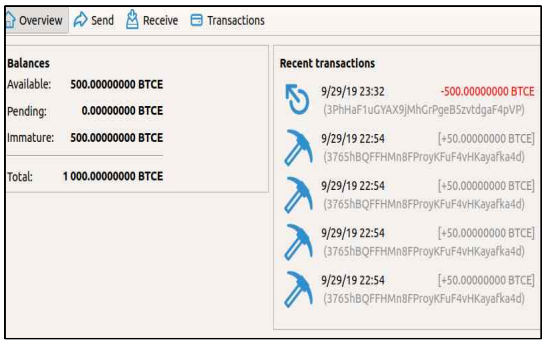


그림 11. node3의 금액과 트랜잭션 기록
Fig. 11. Balances and transaction logs in node 3

더 나아가 그림 11에서는 node 3의 코인 양과 최근 트랜잭션 기록을 확인하여 node 3이 보유하고 있는 코인 양이 감소하였음을 확인할 수 있고 이는 트랜잭션 기록에 기재되었다. 이때, 아직 발생한 트랜잭션을 가지고 있는 블록이 확인되지 않아 코인 총량에는 적용되지 않고 “Immature”에 기재되었다. 다른 node에서 블록을 채굴하여 어느 길이 이상이 되면 적용된다.

V. 평 가

아마존닷컴에서 개발한 클라우드 컴퓨팅 플랫폼(AWS)을 이용하여 데이터수집을 위해 사용하였다. 표 7을 통해 평가환경 세부 항목을 나타내고 있다. Monitoring PC를 통해 ‘오픈 네트워크 테스트’ 결과 확인, Seed Instance는 노드 간의 블록체인 네트워크 연결, Mining Instance는 블록 채굴하는 역할로 나누어 평가를 진행했다. Instance를 생성하기 위해 AMI(Amazon Machine Image), Instance 유형, Instance 구성, 스토리지, 보안 그룹 구성을 선택하여 환경 구축 가능하다는 장점이 있다.

표 7. 평가의 구현환경
Table 7. Implementation environment of evaluation

No	Role	CPU	Memory (GB)	HDD/SSD(GB)	Volume
1	Monitoring PC	Inter(R) Core(R) CPU i5-7600U@ 2.60Hz	16	SSD 256	1
2	Seed Instance	AWS m5.xlarge vCPU 2	8	HDD 20	6
3	Mining Instance	AWS m5.xlarge vCPU 2	8	HDD 20	40

본 평가환경은 Ubuntu Server 18.04 LTS, m5.large (vCPU processor 2 core, RAM 8GB), SSD 20GB 선택하여 진행하였다.

블록체인을 구성하는 요소에서 트레이드 오프(Trade-Off)가 되는 문제가 발생하게 되는 데 이를 트릴레마(Trilemma) 문제라고 한다. 트릴레마의 요소는 분산성(Decentralization), 확장성(Scalability) 그리고 보안성(Security)으로 나뉜다. 본 평가에서는 분산성 평가, 보안성 평가 그리고 확장성 평가를 통해 평가 수치를 계산하고 ECCPoW가 트릴레마 문제를 얼마나 해결되었는지 보여주고자 한다. 표 8은 블록체인 트릴레마 문제를 기준으로 평가하기 위한 항목표이다.

표 8. 평가 항목 기준
Table 8. Description of evaluation features

Evaluation features	Unit of measurement	Evaluation standards	Evaluation goal
Decentralization	Distribution of mining success rate	40% (Estimate) (Bitcoin, Oct.~ Dec. 2018)	40%
Security	Security of Bitcoin contrast	100% (Bitcoin)	100%
Scalability	Scalability of Bitcoin contrast	100% (Bitcoin)	100%

현재 비트코인은 해시 레이트가 매우 높다. 즉, 블록체인 크기가 매우 크다는 뜻이다. ECCPoW를 같은 환경을 만들기 어려움이 있다. 그래서 같은 평가환경(난이도 변경 주기, 목표 블록 생성시간, 23개의 Instance 등)을 설정하기 위해 초기화한 비트코인과 ECCPoW를 비교하였다. 아래의 평가에서 얻은 결과는 난이도 변경 주기 60분, 목표 블록 생성시간 3분으로 설정하였다.

5.1 분산성 평가

분산성이란 네트워크가 중앙집중화를 벗어나 블록체인 내에서 자율적으로 운영되는 것을 말한다. 즉, 참여율과 유사하다. 분산성이 높다는 것은 사용자가 블록체인 참여율이 높다는 것으로 이는 참여에 대한 보상이 잘 이루어지고 있다고 판단할 수 있다.

일반적으로 참여에 대한 보상은 채굴을 통해 이

루어진다. 분산성 평가에 사용된 지표는 채굴성공률 분포도이며, 식 (5)로 정의한다. 식 (5)를 보면, 채굴 성공확률의 분산이 낮을수록 채굴 성공률 분포도가 높아지게 된다. 즉, 참여 노드들 각자의 채굴 성공률이 고르게 분포되어 있을수록 채굴성공률 분포도가 높다. 이는 분산성이 우수하다는 것을 의미한다. 비트코인의 경우 채굴 성공률 분포도가 40%로 추정된다. (기준, 2018년 10~12월) ECCPoW의 분산성 수치도 40%를 목표하고 있다.

$$A = \frac{C}{\sqrt{B + C^2}} \times 100 \quad (5)$$

A : 채굴성공률분포도(%)
B : 채굴성공확률분산
C : 채굴성공확률평균

ECCPoW 블록체인을 구성하기 위해 시드 노드 3개를 생성하고 블록 채굴을 위해 채굴 노드 20개로 구성했다. 100개의 블록이 채굴된 시점에서 각 채굴 노드의 채굴 성공 개수를 확인한 값이 표 9, 이를 이용하여 채굴 성공 분포도를 계산한 값들을 표 10에서 확인할 수 있다.

표 9. 노드별 채굴 성공 개수
Table 9. Success mining number of nodes

Number of mining nodes	Number of mining Success	Number of mining nodes	Number of mining success
1	4	11	7
2	9	12	4
3	3	13	12
4	4	14	5
5	6	15	11
6	4	16	2
7	6	17	7
8	6	18	6
9	5	19	10
10	5	20	8

표 10. 분산성 평가결과
Table 10. Evaluation result of decentralization

Total number of mining success	Average of mining success	Square of average number	Dispersion of average number	Distribution of mining success
124	6.2	38.44	6.76	92.21944

ECCPoW의 분산성은 채굴 성공률 분포도로 측정하였으며 평가 목표 수치인 60%보다 32% 높게 92.22%(소수점 이하 셋째 자리 반올림)로 측정되었다. 이러한 실험을 통해 ECCPoW 블록체인의 참여자는 블록체인에 참여를 통해 받아야 할 보상이 제대로 이루어지는 것 즉, ECCPoW가 채굴 집중화에 강하다고 판단할 수 있다.

5.2 보안성 평가

보안성이 우수함은 블록체인 내의 데이터를 권한이 없는 이용자가 사용하기 어렵다는 것을 뜻한다. 이에 대한 대표적인 문제로써 이중 지불 문제가 있다. 이중 지불 문제의 원인 중 하나는 블록체인의 분기이다. 분기가 생겨서 체인이 형성되는데 이를 고아 체인, 이러한 고아 체인에 속한 블록을 고아 블록이라고 한다. 즉, 블록체인에서 전체 블록의 개수 대비 고아 블록 비율이 낮다면 블록체인의 분기는 적다고 판단할 수 있으며 이를 통해 보안성을 평가할 수 있다.

보안성 평가에서는 식 (6)을 이용하여 비트코인과 ECCPoW의 고아 블록 비율을 계산하고 두 값을 비교한다. 비트코인의 보안성을 100%로 기준을 두어 ECCPoW의 보안성 평가를 했다.

$$\text{고아 블록 비율}(\%) = \frac{(\text{고아 블록 개수})}{(\text{블록체인의 총 높이})} \times 100 \quad (6)$$

비트코인과 ECCPoW 블록체인을 구성하기 위해 각 3개의 시드 노드를 생성하고 블록 채굴을 위해 각 20개의 채굴 노드를 구성한다. 그리고 100개의 블록이 채굴된 시점에서 채굴 종료를 하고 그림 12를 참고하여 '고아 블록' 항목을 확인하여 개수를 파악한다. 이때, 블록체인의 안정적인 동기화를 위해 블록 높이가 40이 된 이후의 블록을 확인하였다. 단, 부 체인을 속한 고아 블록의 개수를 파악하는데 높이 1은 두 환경 모두 발생 빈도가 높아 보안성 평가에 적절하지 못하다. 즉, 체인의 유효성이 높은 높이 2 이상의 체인으로 형성되어야 고아 블록이라고 판단하였다.

그림 12의 블록 13, 블록 14, 블록 15, 블록 16, 블록 18이 해당한다. 이렇게 파악한 고아 블록의 개수를 고아 블록 비율 구하는 식을 이용하여 비트코인과 ECCPoW의 고아 블록 비율 비교를 했다.

그림 13는 비트코인 환경에서의 보안성 평가 결과에 따른 블록체인 상태를 보여주고 있으며 그림 14은 ECCPoW 환경에서의 보안성 평가 결과에 따른 블록체인 상태를 보여주고 있다. 표 11에서는 고아 체인 블록 개수, 고아 체인 블록 비율을 확인할 수 있다. 모두 0%로 측정되었으므로 비트코인의 보안성이 100%라는 기준으로 ECCPoW의 보안성도 같은 결과를 도출할 수 있었다.

표 11. 보안성 평가결과
Table 11. Evaluation result of security

Blockchain name	Number of orphan blockchain	Ratio of orphan blockchain	Security
Bitcoin	0	0	100
ECCPoW	0	0	100

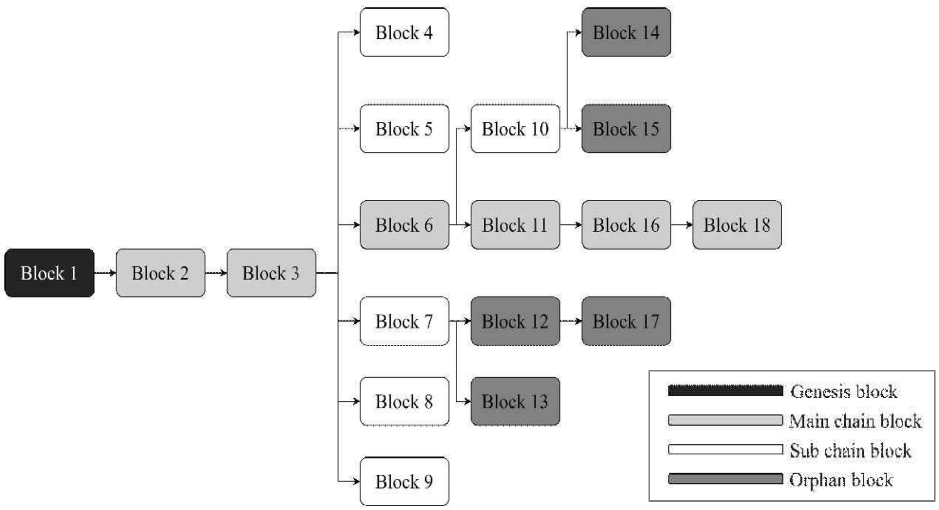


그림 12. 보안성 평가 설명도
Fig. 12. Diagram description of security evaluation

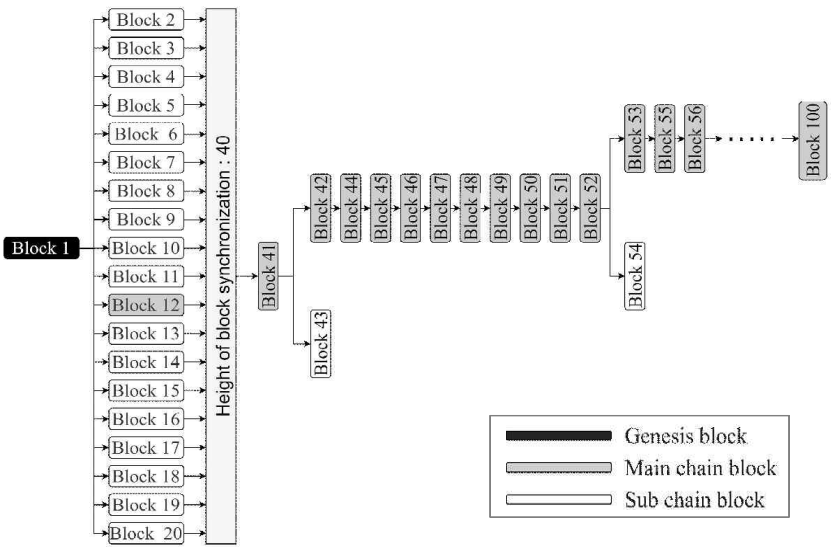


그림 13. 비트코인 보안성 평가도
Fig. 13. Evaluation diagram of Bitcoin security

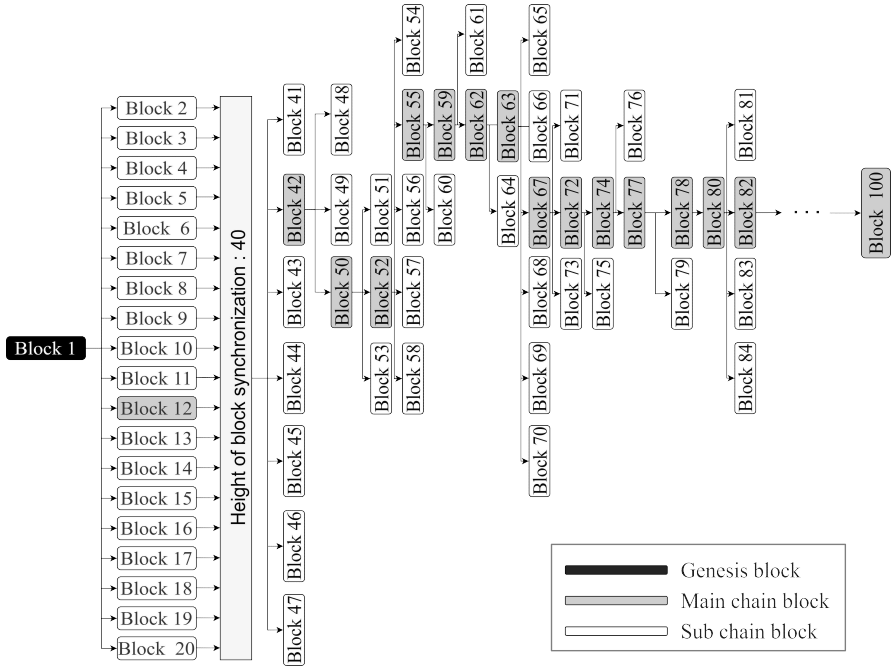


그림 14. ECCPoW 보안성 평가도
Fig. 14. Evaluation diagram of ECCPoW security

5.3 확장성 평가

블록체인 확장성이란 블록체인의 서비스 수용성을 의미한다. 즉, 서비스를 원활하게 이뤄지게 하려면 서비스의 속도가 중요한데 결국 TPS(Transaction Per Second)로 직결된다. TPS는 초당 트랜잭션 처리 지표로써 블록체인의 처리 속도를 의미하고 있다. TPS가 높으면 확장성이 좋다고 판단할 수 있다. 비트코인과 ECCPoW의 각 TPS를 측정하여 비교함으로써 ECCPoW의 분산성을 평가할 수 있다. 식 (7)은 10개의 블록이 담긴 트랜잭션을 이용한 TPS를 구하는 식이다.

$$TPS = \frac{(10\text{개 블록의 총 보유 트랜잭션 수})}{(10\text{개 블록의 총 생성 시간})} \quad (7)$$

비트코인과 ECCPoW 블록체인을 구성하기 위해 각 3개의 시드 노드를 생성하고 블록 채굴을 위해 각 20개의 채굴 노드를 구성한다. 트랜잭션 발생은 높이 91부터 높이 100까지 지속해서 발생시키고 100개의 블록이 채굴된 시점에서 채굴이 종료한다. 각 블록체인 91~100의 높이에 해당하는 각 블록의 보유 트랜잭션 양을 확인하여 TPS 식을 이용하여 TPS를 구하고 비교 평가를 했다.

표 12에서는 비트코인 높이 91~100의 블록의 생성 소요시간과 보유 트랜잭션 수, 표 13에서는 ECCPoW 높이 91~100의 블록의 생성 소요시간과 보유 트랜잭션 수를 확인할 수 있다.

표 12. 비트코인 확장성 평가결과
Table 12. Evaluation result of Bitcoin scalability

Number	Height	Block generation time (seconds)	Number of transactions
1	91	719	672
2	92	19	22
3	93	94	88
4	94	144	141
5	95	275	263
6	96	40	40
7	97	313	296
8	98	6	11
9	99	574	534
10	100	146	137
Total		2330	2204
TPS		0.945922747 TPS	

표 13. ECCPoW 확장성 평가결과
Table 13. Evaluation result of ECCPoW scalability

Number	Height	Block generation time (seconds)	Number of transactions
1	91	151	140
2	92	192	182
3	93	369	351
4	94	361	331
5	95	156	151
6	96	214	205
7	97	124	120
8	98	267	250
9	99	585	548
10	100	514	480
Total		2933	2758
TPS		0.940334129 TPS	

이를 통해 비트코인의 TPS는 0.95(소수점 이하 셋째 자리), ECCPoW의 평균 TPS는 0.94(소수점 이하 셋째 자리 반올림)로 비트코인의 확장성이 100%라는 기준으로 ECCPoW의 확장성은 1.02% 감소한 98.95%로 측정되었다.

Ⅵ. 결 론

이 논문에서는 ECCPoW에 대하여 설명하고 비트코인에 제안방법을 적용하였다. ECCPoW는 ASIC 저항성을 확보하기 위하여 매 블록 다른 문제를 푸는 방법을 제안하였다. 이는 기존연구들이 한정된 몇 가지 해시 함수를 연결하여 사용하는 방법의 장점을 극대화하여 매 블록 다른 해시 함수를 푸는 효과를 보여준다.

우리는 ECCPoW를 구현하기 위한 난이도 조절, 패리티 체크 행렬 생성방법, 해시 벡터 생성 및 codeword 판별 방법을 제시하였다. 그리고 이를 검증하기 위하여 비트코인에 ECCPoW를 적용하였다. 비트코인과 분산성, 보안성, 확장성 측면에서 비교 평가하였다. ECCPoW는 보안성과 확장성을 유지한 상황에서 비트코인보다 32% 높은 분산성을 보였다. 이를 통하여 ECCPoW는 높은 해시 레이트를 요구하지 않으며 채굴자들은 더욱 형평에 맞은 경쟁이 가능하다.

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", 2009.

[2] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain", Business & Information Systems Engineering, Vol. 58, pp. 183-187, Mar. 2017.

[3] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen, "Blockchain technology and its relationships to sustainable supply chain management", International Journal of Production Research, Vol. 57, No. 7, pp. 2117-2135, Oct. 2018.

[4] R. Qin, Y. Yuan, and F. Y. Wang, "Research on the selection strategies of blockchain mining pools", IEEE Transactions on computational social systems, Vol. 5, No. 3, pp. 748-757, Sep. 2018.

[5] X. Liu, W. Wang, D. Niyato, N. Zhao, and P. Wang, "Evolutionary game for mining pool selection in blockchain networks", IEEE Wireless communications letters, Vol. 7, No. 5, pp. 760-763, Oct. 2018.

[6] G. O. Karame, E. Androulaki, and S. Capkun, "Double-Spending Fast Payments in Bitcoin", In Proceedings of the 2012 ACM conference on Computer and communications security, pp. 906-917, 2012.

[7] J. H. Jang and H. N. Lee, "50% less then double-spend attack", OSIA S&TR Journal, Vol. 32, No. 1, pp. 4-10, Mar. 2019.

[8] Antminer S19 pro, https://support.bitmain.com/hc/en-us/articles/900000261726-S19-Pro-Specifications. [accessed: Apr. 20. 2020]

[9] S. Park, H. Kim, and H. N. Lee, "Introduction to Error-Correction Codes Proof-of-WorK", The Magazine of the IEIE, Vol. 5, No.46, pp. 26-32, May 2019.

[10] V. Buterin, "A next-generation smart contract and decentralized application platform", white paper, 2014.

[11] G. Wood, "ETHEREUM: A SECURE

DECENTRALISED GENERALISED SRANSACTION BYZANTIUM VERSION", yellow paper, 2014.
[12] Q. Zhou, H. Huang, Z. Zheng and J. Bian, "Solutions to Scalability of Blockchain: A Survey", IEEE Access, Vol. 8, pp. 16440-16455, Jan. 2020.
[13] IfDefElse and G. Colvin, "ProgPoW, a Programmatic Proof-of-Work", Ethereum - EIPs, No. 1057, May 2018.
[14] E. Duffield and D. Diaz, "Dash: A PrivacyCentric Crypto-Currency", white paper, Aug. 2018.
[15] N. V. Saberhagen, "Cryptonote v2.0", white paper, Oct. 2013.
[16] RandomX, https://github.com/tevador/RandomX/blob/master/doc/specs.md. [accessed: Apr. 20. 2020]
[17] R. G. Gallager, "Low Density Parity Check Codes", IRE Transactions on Information Theory, Vol. 8, No. 1, pp. 21-28, Jan. 1962.
[18] https://github.com/cryptoecc/bitcoin_ECC/blob/ecc-0.1/src/ldpc/LDPC.cpp. [accessed: Apr. 20. 2020]
[19] Y. Ben-Haim and S. Litsyn, "Upper bounds on the rate of LDPC codes as a function of minimum distance", IEEE Transactions on Information Theory, Vol. 52, No. 5, pp. 2092-2100, May 2006.

저자소개

정 현 준 (Hyunjun Jung)

2008년 : 삼육대학교 컴퓨터과학과(학사)
2010년 : 숭실대학교 컴퓨터학과(공학석사)
2017년 : 고려대학교 컴퓨터·전파통신공학과(공학박사)
2017년 8월 ~ 현재 : 광주과학기술원 블록체인인터넷경제연구센터
관심분야 : 블록체인, 데이터 사이언스, 센서 네트워크, 사물인터넷

채 종 홍 (Jong-Hong Chae)

2019년 : 조선대학교 정보통신공학과(학사)
2019년 3월 ~ 2020년 2월 : 광주과학기술원 블록체인인터넷경제연구센터 연구원
관심분야 : 정보통신, 인공지능, 블록체인

이 흥 노 (Heung-No Lee)

1993년 : University of California 전기공학과 졸업
1994년 : University of California 전기공학과 석사
1999년 : University of California 전기공학과 박사
1999년 ~ 2002년 : HRL Laboratories Research Staff Member
2002년 ~ 2008년 : University of Pittsburgh Assistant Professor
2009년 ~ 현재 : 광주과학기술원 전기전자컴퓨터공학부 교수
관심분야 : 정보이론, 신호처리, 통신/네트워크, 압축센싱, 블록체인, 센서지능화