Journal of KIIT. Vol. 18, No. 6, pp. 33-42, Jun. 30, 2020. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2020.18.6.33

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 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 채굴 중앙화 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 mining centralization.

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 전기전자컴퓨터공학부 교수(교신저자)
- ORCID: https://orcid.org/0000-0001-8528-5778

· Received: Feb. 19, 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

블록체인의 등장은 우리에게 새로운 신뢰를 보장할 기회를 제공한다. 인터넷에서 거래에서 신분 증명은 제3의 신뢰 기관에 의존하여 신뢰를 보장받는다. 하지만 신뢰 기관은 사람에 의해 운영되어 사람에 의한 도덕적 위험을 포함하고 있다. 신뢰 보증은 대체할 수 있는 기술이 없으므로 법적인 제재를 이용하여 실시한다. 비트코인(Bitcoin)은 P2P 네트워크를 이용하여 신뢰 기관 없이 운영되는 전자 화폐 시스템을 제안했다[1].

비트코인에서 사용된 블록체인은 암호화 함수를 실행하는 해싱(Hashing)작업을 이용하여 블록을 생성한다. 블록체인을 유지하고 있는 P2P 노드들은 모든 거래명세가 들어있는 블록을 저장하여 변경되지 않았음을 서로 증명한다. 블록체인은 기존의 제3의 기관에 의존한 신뢰 증명을 기술에 의한 증명으로 패러다임의 전환이다. 블록체인의 노드들은 해시 기반 작업증명(PoW, Proof-of-Work)을 이용하여 블록의 내용이 변경되지 않았음을 증명한다. 블록은 생성될 시 타임스탬프(Timestamp)를 이용하여 특정 시간에 존재하고 있었음을 증명한다. 노드는 블록을 생성하기 위하여 해시 함수 연산을 수행하고, 가장 빨리 증명한 노드에 보상을 준다. 그 결과, 블록체인은 체인형대로 연결된 블록을 네트워크에 참여한 노드들이 협력하여 변경되지 않음을 증명한다. 결국, 블록체인은 누구라도 임의로 수정할 수 없는 분산 컴퓨팅 기술기반의 원장 관리를 가능하게 한다[2][3].

해시 레이트(Hash rate)는 암호화폐를 채굴하기 위한 연산 처리 능력을 측정하는 단위로 초당 해시 값 계산 횟수를 말한다. 암호화폐는 일정한 블록생성 시간을 목표로 난이도를 조절한다. 암호화폐에 참여하는 노드가 많아지거나, 참여한 노드들의 컴퓨팅 성능이 좋아지면 목표한 블록생성시간을 위하여 해시 레이트는 점점 증가한다. 해시 레이트가 낮을 경우에는 이중 지불공격에 대한 위험이 있다[4]. 또는 소수의 그룹이 전체 해시 레이트의 51% 이상을 점유한다면 블록체인을 자신이 원하는 쪽으로 악용할 위험이 생긴다. 해시 레이트가 높을 경우에는 블록을 생성하기 위하여 많은 연산량이 필요하다. 블록체인의 해시 레이트가 높아질수록 블록이 변경에 대한 안정성은 올라가지만 소모되는 컴퓨팅 파워(전기)가 많아진다는 단점도 유발하게 된다. 그림 1은 비트코인 해시 레이트 변화도를 보여준다. 비트코인은 블록체인의 규모가 커지고 참여하는 노드의 수가 증가함에 따라 해시 레이트가 증가하였다. 해시 레이트가 증가함에 따라 개인은 채굴기(Minning machine)를 이용하거나 채굴 풀(Minning pool)에 소속하여 채굴한다[5][6].

채굴기는 해시 레이트의 변화에 따라 개인 컴퓨터의 중앙처리장치(CPU)를 이용하던 시기에서 그래픽 카드에 사용되는 CPU를 이용하던 시기를 거쳐 채굴 전용 주문형 반도체(ASIC, Application Specific Integrated Circuit)를 사용한다. ASIC 채굴기는 단순 해시 함수만을 위해 단순 반복 연산을 빠르게 처리할 수 있다. ASIC 채굴기는 일반연산용인 CPU나 GPU보다 채굴성능이 월등히 뛰어나다. 비트메인의 채굴기 Antminer S9(13,000,000MH/s)은 GPU GTX1060(1478MH/s)과 비교했을 때 약 8,800배 차이가 난다. CPU/GPU를 사용하는 채굴자와 ASIC 칩을 사용하는 채굴자는 채굴에 성공할 확률이 8,800배 난다고 말할 수 있어서 형평에 어긋난다. 비트코인은 시간이 지날수록 ASIC 채굴기를 사용한 채굴자에게 화폐가 집중될 것이다.

PoW 블록체인은 해시 레이트가 커질수록 블록생성에 드는 컴퓨팅 파워가 많아진다. 일부 사람들은 PoW 블록체인의 블록생성을 위해서 낭비되는 많은 전력을 문제 삼는다. 하지만 PoW 블록체인은 블록생성에 드는 전력으로 높은 안정성을 보증한다. PoW 블록체인은 ASIC 채굴기로 인하여 개발을 억제하기 위한 많은 연구가 진행되었다.

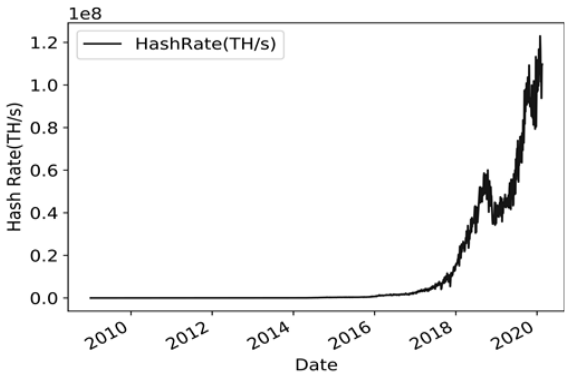

그림 1. 비트코인 해시 레이트 변화
Fig. 1. Change in bitcoin hash rate

우리는 ASIC 채굴기의 개발을 억제할 수 있는 새로운 채굴 함수로써, LDPC(Low Density Parity Check) 디코더와 해시 함수를 결합한 오류-정정 부호 기반의 작업증명(ECCPoW Error-Correction Codes Proof-of-Work)를 제안하였다[7].

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 ECCPoW와 ASIC 채굴기의 개발을 방지하기 위해 시도된 연구를 소개한다. 3장에서는 ECCPoW의 구현방법에 대하여 제시한다. 4장에서는 ECCPoW를 비트코인에 탑재하여 실험한 결과를 소개한다. 5장에서는 구현한 ECCPoW의 성능을 채굴 중앙화로 평가한다. 마지막으로 6장에서는 결론을 제시한다.

II. 관련 연구

2.1 오류-정정 부호 기반 작업증명(EEPoW)

우리는 AISC 저항성을 높이기 위한 ECCPoW의 개념을 제안하였다[9]. ECCPoW는 통신에서 많이 사용되는 오류-정정 부호(Error-correction codes)의 디코더를 활용한 작업증명이다. 오류-정정 부호에서 사용되는 디코더는 일반적으로 ASIC 장치를 이용해 구현할 수 있다. 오류-정정 부호를 위한 디코더의 설계는 패리티 체크 행렬(H)에 의해 결정된다. 즉 패리티 체크 행렬을 고정하면 디코더를 ASIC로 제작할 수 있다. 핸드폰과 같이 표준화된 패리티 체크 행렬이 결정돼있으면 오류-정정 부호 디코더 ASIC 설계가 가능하다. 하지만, 무수히 많은 패리티 체크 행렬들을 지원하는 디코더에 맞게 ASIC를 제작하는 것은 비용 문제 및 디코더의 크기 문제로 어렵다. ECCPoW는 매 블록 패리티 체크 행렬이 무작위로 변한다. 다시 말해, ECCPoW는 사용하는 패리티 체크 행렬의 개수가 무한하다. 그 결과, ECCPoW는 오류-정정 부호 디코더를 위한 ASIC 개발을 억제한다. ECCPoW의 디코딩 알고리즘은 CPU 혹은 GPU로만 실행하게 된다.

2.2 PoW 블록체인의 ASIC 저항성 연구

PoW 블록체인을 위한 ASIC 저항성 연구는 세 가지 접근방법이 있다. 첫 번째 방법은 해시 함수에서 메모리의 접근을 강제를 이용하여 병목현상을 유발한다. 두 번째 방법은 해시 함수의 중첩을 이용하여 ASIC의 생성을 저지한다. 세 번째 방법은 주기적으로 해시 함수를 주기적으로 교체하는 방법이다.

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[8]. 이더리움은 비선형 그래프(DAG, Directed Acyclic Graph)를 이용하여 ASIC에 대항한다. DAG는 해시 함수가 임의의 메모리를 반복적으로 참조하도록 알고리즘을 설계하여 캐시 미스를 유도한다. DAG의 초기 크기는 약 1GB이었으며, 천천히 시간이 지날수록 선형으로 크기가 증가하도록 설계되었다. 2019년 10월 현재 DAG의 크기는 3.99GB이며 2020년 12월 20일까지 유지된다[9][10]. 이더리움은 2019년 ASIC의 채굴 중앙화에 대응하기 위하여 ProgPoW [13]를 개발하여 적용하기로 승인하였다.

X-11은 뒤에 붙은 숫자만큼의 해시 함수를 사용하는 암호화폐 채굴 알고리즘이다[11]. X11은 ASIC를 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.

크립토노트를 적용한 모네로는 이를 막기 위해 일 년에 두 번씩 채굴 알고리즘을 변경하는 방안을 시행 중이다[12]. 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[13].

III. ECCPoW Blockchain 구현방법

이 장에서는 우리가 제안한 ECCPoW의 구현방법에 대하여 보여준다. 우리는 ASIC 저항성을 높이기 위하여 ECCPoW의 개념을 제안하였다. 기존 ASIC 저항성 연구는 메모리의 적재를 유도하는 방법, 여러 해시 알고리즘을 사용하는 방법, 일정한 기간을 주기로 해시 함수를 변경하는 방법이 있다. 여러 해시 알고리즘을 중첩해서 사용하는 방법은 ASIC 채굴기가 등장하였다. 또한, 메모리 적재 방법은 낮은 성능의 ASIC가 발매되었다. 일정 주기로 해시 알고리즘을 변경하는 방법은 잦은 하드포크로 사용자의 이용에 불편을 만들었다.

ECCPoW는 ASIC 저항성을 극대화하기 위해 블록마다 바뀌는 암호 퍼즐 생성을 제안한다. 이 방법은 주기적으로 해시 함수를 변경하는 방법의 단점을 줄이며 ASIC 채굴기의 등장을 억제할 것으로 기대한다.

3.1 매 블록 바뀌는 암호 퍼즐 생성

ECCPoW은 매 블록 다른 암호 퍼즐을 생성하여 ASIC 저항성을 확보하려 한다. 그림 2는 ECCPoW 블록체인의 블록생성 개요를 보여준다. ECCPoW의 핵심은 디코더의 입력같인 패리티 체크 행렬(H)이 변경되어 채굴자들이 매 블록 다른 암호 퍼즐 푸는 효과이다. 패리티 체크 행렬은 이전 블록의 내용을 기준으로 생성한 seed를 이용하여 만든다. seed는 이전 블록의 내용의 일부 중 매 블록 변경되는 요소(Previous block hash, merkle root 등)의 요소로 생성한다. 패리티 체크 행렬은 seed 값의 요소가 조금만 변해도 완전히 내용이 재구성되는 특징을 가지고 있다. 생성된 패리티 체크 행렬은 현재 블록을 생성하는 동안에는 고정된다. 즉, 모든 채굴자는 현재 블록을 생성할 때 같은 암호 퍼즐을 푼다. 채굴자들은 nonce를 생성하고 SHA 함수 등을 이용하여 해시 벡터(e)를 생성한다. 그리고 채굴자는 디코더에 패리티 체크 행렬과 해시 벡터를 입력값으로 실행한다. ECCPoW 블록체인은 디코더의 실행결과인 ouput word를 확인하고, 블록생성 조건에 해당하는지 확인 후 블록을 생성한다.

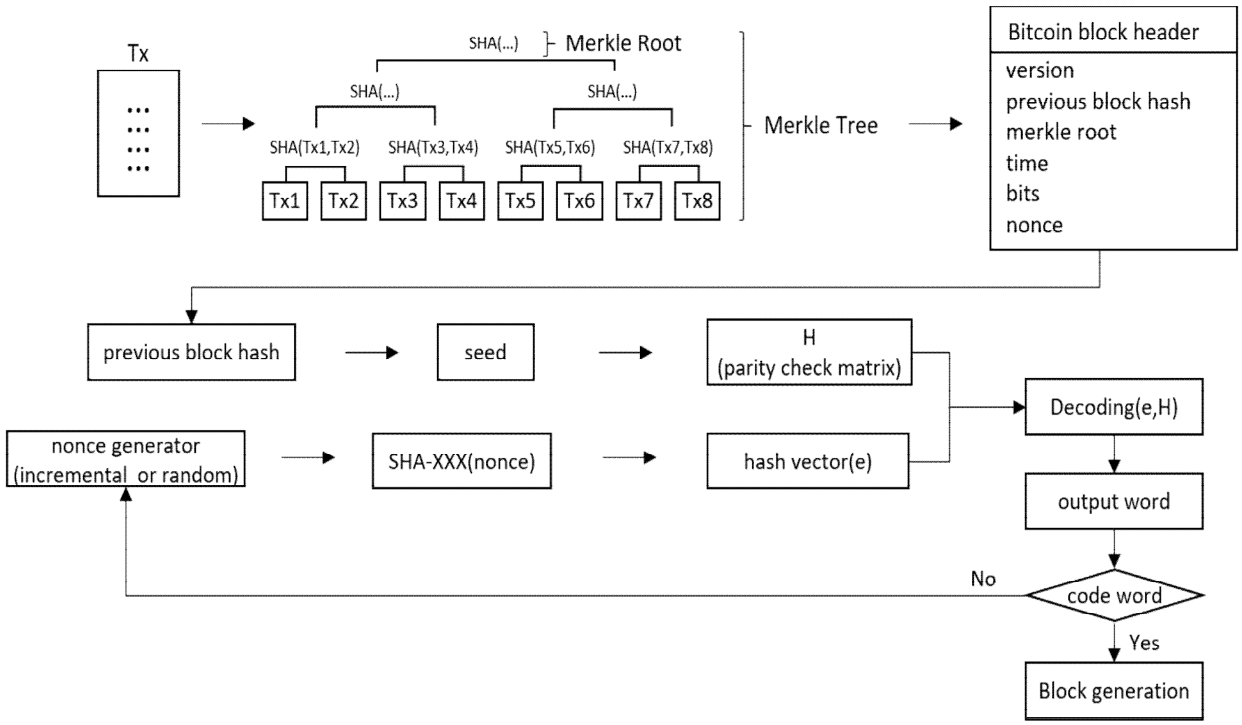

그림 2. ECCPoW 블록체인의 블록생성 개요
Fig. 2. Overview of block generation of ECCPoW blockchain

이 논문에서는 Gallager[14]의 생성방법과 이전 해시값을 동시에 사용해 암호 퍼즐 생성에 사용되는 합성 함수를 매 블록 변경하였다. ECCPoW는 순열 순서를 이전 해시값을 통해 변경하게 된다. 이전 해시값을 시드 값으로써 활용하여 순열 순서를 결정한다. 해시값은 무작위의 값이므로 순열 순서를 무작위가 된다. [15]에서는 구현한 코드를 확인할 수 있다.

표 1은 서로 다른 이전 해시값을 사용 시, 생성된 H를 비교하였다. Gallager의 방법을 이용하기 위해서는 다음의 변수가 필요하다. Gallager의 변수 n은 H 열의 개수, m은 H의 행의 개수, w_c는 H 열의 1의 개수, w_r은 H각 행의 1의 개수를 의미한다.

표 1. 서로 다른 해시값을 사용했을 때의 생성된 H의 형태
Table 1. Form of the resulting H of using a different hash value

Input value	Part of Generated H
n=12 m=24 w_c=3 w_r=6 seed=1919	1 1 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0 1 0 0 1 0 0 0 0 1 0
n=12 m=24 w_c=3 w_r=6 seed=2327	1 1 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0

3.2 매 블록 바뀌는 암호 퍼즐 디코더

ECCPoW는 매 블록 생성된 암호 퍼즐을 풀기 위한 디코더가 필요하다. 디코더는 해시의 출력값과 생성된 패리티 체크 행렬을 입력값으로 취득한다. 그리고 메시지 전달 알고리즘(Massage-passing algorithm)에 기초한 디코딩을 수행한다. 그 결과를 출력값으로 산출하고 암호 퍼즐의 해결 여부를 판단한다.

ECCPoW의 LDPC 디코더는 메시지 전달 기반 알고리즘을 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.

ECCPoW는 암호 퍼즐 해결 여부를 판단하는 두 가지 기준을 사용한다. 첫 번째, 디코더의 결괏값이 부호이고, 특정 해밍 가중치를 가지면 해결한 것으로 판단한다. 두 번째, 디코더의 결괏값을 다시 해싱하여 나온 값이 특정 값보다 작으면 해결한 것으로 판단한다.

첫 번째 기준은 디코더의 출력값 c가 조건을 만족시키면 암호 퍼즐을 해결한 것으로 판단한다. 조건 1은 출력값이 부호, 조건 2는 출력값의 해밍 가중치가 원하는 집합의 원소인 것이다. 이 조건은 디코더가 임의의 입력값을 받았을 때, 부호를 산출 할 확률이 적다는 것에서 기인한다. 그리고 패리티 체크 행렬이 주어졌을 때 생성 가능한 부호들의 해밍 가중치들이 다를 수 있다는 것에서 기인한다.

조건 1은 풀 저항성 확보와 연관되어 있다. 마이닝 풀이 등장하면, 풀의 구성원들은 암호 퍼즐을 풀기 위해 노력하고 있다는 것을 풀의 관리자에게 증명한다. 이를 위해서는 부분정답이라는 것을 제출한다. 예를 들어 비트코인 채굴 풀의 경우 0이 20개로 시작하는 정답을 찾는 암호 퍼즐을 풀 때, 0이 6개로 시작하는 해답을 부분정답으로 제출하기로 설정할 수 있다. 부분정답 중에는 올바른 정답이 포함되어 있으므로, 풀 관리자는 부분정답을 제출받는 것으로 풀 구성원들이 퍼즐을 풀고 있다는 사실을 확인함과 동시에 제출받은 부분정답 중에 포함된 올바른 정답을 사용하여 채굴할 수 있다. 하지만 ECCPoW에서 조건 1을 사용한다면 효과적인 부분정답 선정이 어렵다. 따라서 풀 관리자로서는 풀 구성원들의 성실함을 확인하기 힘들어서 마이닝 풀을 운영하기 위한 동기가 부족해진다.

조건 1을 만족할 확률을 구하려면 의 최소 해밍 거리 값이 필요하다. 이 값을 계산하려면 2k의 서로 다른 부호들을 모두 고려해야 한다. 부호의 개수가 적을 때에는 가능하지만 개수가 클 때는 불가능하다. Litsyn[19]은 특정 w_c, w_r일 때, 패리티 체크 행렬의 최소 해밍 거릿값의 상한/하한값들을 보고하였다. 조건 2는 변수들 n, m, w_c, w_r이 고정되었을 때, 암호 퍼즐의 난이도를 높이기 위하여 사용된다.

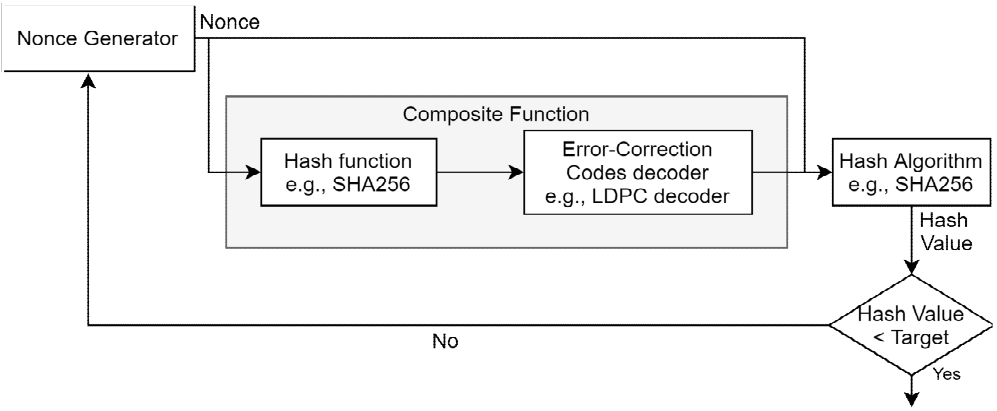

그림 3. ECCPoW 암호 퍼즐 해결 유무 판단 기준 2
Fig. 3. Criteria 2 for ECCPOW crypto puzzle resolution determination

두 번째 기준은 디코더의 출력값과 논스를 다시 해싱하여 나온 결괏값을 얻고 해당 결괏값이 정해진 대상(Target)과 비교하여 암호 퍼즐 해결 여부를 판단한다. 그림 3은 ECCPoW 암호 퍼즐 해결 여부 판단 과정을 보여준다. 합성 함수와 해시 알고리즘을 하나의 해시 함수로 인식하면 비트코인과 같은 구조이므로, 비트코인의 난이도 조절 함수를 사용할 수 있다.

Ⅳ. 실 험

이 장에서는 3장에서 제안한 ECCPoW를 구현하여 실험한 결과를 보여준다. 단일 노드 실험을 통하여 비트코인의 합의 알고리즘을 ECCPoW로 교체한 블록생성 기능을 확인한다.

ECCPoW의 핵심 모듈들은 C++ 언로로 구현되어 있다. 표 2는 각 모듈과 관련된 함수들 함수 리스트를 보여준다.

ECCPoW를 실험하기 위해서는 몇 가지 파라미터의 설정이 필요하다. Chanparams.cpp 파일의 내용에서 실험을 위한 파라미터를 설정할 수 있다[16]. 표 3은 ECCPoW와 관련된 주요 파라미터 설명이다.

그림 4는 ECCPoW의 동작 과정을 보인다. 기본 동작을 보여주기 위하여 비트코인의 RPC 명령어를 사용하였다. 실험은 기본적인 계정을 만들어서 새로운 블록을 만든다. 그리고 새로운 블록을 생성을 기다린다. 그리고 생성된 블록의 정보와 계좌에 들어온 암호화폐를 확인한다.

표 2. 각 모듈별 관련 함수 리스트
Table 2. List of related functions for each module

Module name	Related function
Crypto puzzle generation	generate_seeds, generate_H, generate_Q
Crypto puzzle decoder	generate_hv, decoding
Crypto puzzle resolution judgment	condition 1: decision, condition 2: binary_to_hex
Crypto puzzle difficulty control	set_difficulty

표 3. ECCPoW 파라미터 설명
Table 3. ECCPoW parameters description

ECCPoW parameters	Description
powLimit	Minimum difficulty setting
nPowTargetTimespan	Difficulty level change cycle (default 60 minutes)
nPowTargetSpacing	Block generation cycle (default 1 minute)
fPowAllowMinDifficultyBlocks	Permission below minimum difficulty level
fPowNoRetargeting	Permission to change difficulty level
nDefaultPort	ECCPoW Blockchain port number (default port 9777)
init_level	ECCPoW difficulty level (currently default level 10)
CreateGenesisBlock	Linux time, nonce, nBit, nVersion, compensation amount entered in order to generation the genesis block
vSeeds.emplace_back	Connection seed address
m_fallback_fee_enabled	Permission of coin transfer fee setting

```
jun@ubuntu:~/bitcoin_ECC$ bitcoind -txindex -daemon

mainnet with level = 1
set is constructed from 10 to 22 with step 2
n : 32   wc : 3  wr : 4
mainnet n: 399 Hash: 0ccc07b781737bf3c901307dd92ed0548215b745b38daee5fdbb4f042dc4885c

testnet with level = 40
set is constructed from 30 to 30 with step 2
n : 60   wc : 3  wr : 4
testnet n: 47471 Hash: 83432832323855c30a104d61467ecb339ffa988cb8abe420c67acb95667bdd42

regtest with level = 30
set is constructed from 18 to 42 with step 2
n : 60   wc : 3  wr : 4
regtest n: 746 Hash: db20aafd0c626db7cf83b4b3d9ba0a024a862fdff203074e06aaa64e31619461

mainnet with level = 1
set is constructed from 10 to 22 with step 2
n : 32   wc : 3  wr : 4
mainnet n: 399 Hash: 0ccc07b781737bf3c901307dd92ed0548215b745b38daee5fdbb4f042dc4885c
Bitcoin server starting

jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli getnewaddress
39naMhHQgXwm7CVdEaL4PGUkp4fUdHX14M

jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli generatetoaddress 10
9naMhHQgXwm7CVdEaL4PGUkp4fUdHX14M
[
  "fa2549dbcbecfbaff696abd19e086b82e7288cc3c726b6fc4cfc3754a4200a31",
  "09df0e57e1a4f84c594a5553dd2c3cc01c380ba92bf37268a84fbad62ed79693",
  "abc2bce6e224b68da443b505d779cf91715119847d457068c47fa0529dbf71a4",
  "951ae98c7193e11f7469245ec2f18a20103e8e26afe3d7eef10e7a22d15ed2dd",
  "c55a5294d3f8b0c3259cf3842ad0ad9bbb3309523938919dbd59cd1cb14b530a",
  "b2d7b23c8c3ba403b05f6111f49c65212eba0ca61378e15612c04269e3177a91",
  "c4ea7cfc8cafe7e9e8989a5c5808d3e01e83794ebd56278ea9013571e480791d",
  "ac8eb56da73ee9bce603cdda83b1a89903e6cd6828ea5f9c6ab0c8572a780f4b",
  "bd1f46d19d6db8c35536594e6a39339c15ef0185bd244afe7c0af022e6ebf29e",
  "5d4cc21450bcb8df125b8a5301e80f2bc35151ffd1d47a65999238073733507e"
]

jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli getblockchaininfo
{
  "chain": "main",
  "blocks": 10,
  "headers": 10,
  "bestblockhash": "5d4cc21450bcb8df125b8a5301e80f2bc35151ffd1d47a65999238073733507e",
  "difficulty": 4.523059468369196e+74,
  "mediantime": 1590478786,
  "verificationprogress": 1,
  "initialblockdownload": false,
  "chainwork": "00000000000000000000000000000000000000000000000000000000000573f8",
  "size_on_disk": 3285,
  "pruned": false,
  "softforks": [
    {
      "id": "bip34",
      "version": 2,
      "reject": {
        "status": false
      }
    },

jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli getbalance
450.00000000
```

그림 4. ECCPoW 실험
Fig. 4. ECCPoW experiment

V. 평 가

아마존닷컴에서 개발한 클라우드 컴퓨팅 플랫폼(AWS)을 이용하여 평가를 수행하였다. 표 4는 평가 환경에 사용한 노드들의 구성을 보여준다. 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. 본 평가환경은 Ubuntu Server 18.04 LTS, m5.large (vCPU processor 2 core, RAM 8GB), SSD 20GB 선택하여 진행하였다.

우리는 채굴 중앙화(Mining centralization)를 기준으로 평가하였다. 채굴 중앙화란 네트워크가 중앙집중화를 벗어나 블록체인 내에서 자율적으로 운영되는 것을 말한다.

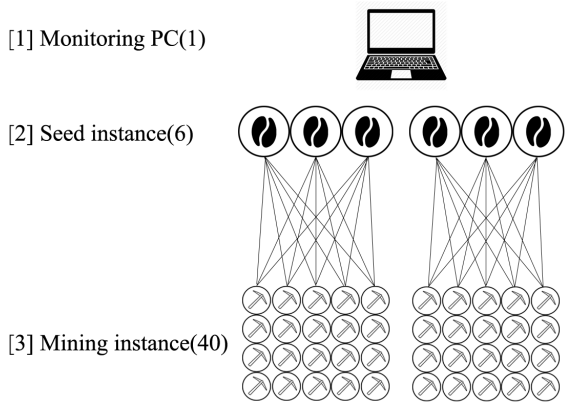

그림 5. 시스템 테스트 환경 구성
Fig. 5. System test environment configuration

표 4. 평가의 구현환경
Table 4. Implementation environment of evaluation

No	Role	CPU	Memory (GB)	HDD/SSD(GB)	Volume
1	Monitoring PC	Inter 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

채굴 중앙화가 높다는 것은 사용자가 블록체인 참여율이 높다는 것으로 이는 참여에 대한 보상이 잘 이루어지고 있다고 판단할 수 있다. 일반적으로 참여에 대한 보상은 채굴을 통해 이루어진다. 채굴 중앙화 평가에 사용된 지표는 채굴 성공률 분포도이며, 식 (1)로 정의한다. 식 (1)을 보면, 채굴 성공확률의 분산이 낮을수록 채굴 성공률 분포도가 높아지게 된다. 즉, 참여 노드들 각자의 채굴 성공률이 고르게 분포되어 있을수록 채굴 성공률 분포도가 높다. 이는 채굴 성공률이 우수하다는 것을 의미한다. 비트코인의 경우 채굴 성공률 분포도가 40%로 추정된다(2018년 10~12월).

$$A = \frac{C}{\sqrt{B + C^2}} \times 100 \quad (1)$$

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 5, 이를 이용하여 채굴 성공 분포도를 계산한 값들을 표 6에서 확인할 수 있다.

표 5. 노드별 채굴 성공 개수
Table 5. 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 6. 채굴 중앙화 평가결과
Table 6. Evaluation result of mining centralization

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 채굴 중앙화는 채굴 성공률 분포도로 측정하였으며 평가 비교 수치인 40% 보다 52% 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.

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 채굴 중앙화 측면에서 비교 평가하였다. 비트코인보다 ECCPoW는 52% 높은 채굴 성공률을 보였다. 이를 통하여 ECCPoW는 높은 해시 레이트를 요구하지 않으며 채굴자들은 더욱 형평에 맞은 경쟁이 가능하다는 것을 확인할 수 있었다.

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", 2009. https://git.dhimmel.com/bitcoin-whitepaper/

[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] 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.

[5] 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.

[6] 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.

[7] 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.

[8] V. Buterin, "A next-generation smart contract and decentralized application platform", white paper, pp. 1-33, 2014. https://arxiv.org/pdf/1511.05740.pdf [accessed: Apr. 20. 2020]

[9] G. Wood, "Ethereum: A Secure Decentralised Generalised Sransaction Byzantium Version", yellow paper, pp. 1-32, 2014. https://gavwood.com/paper.pdf. [accessed: Apr. 20. 2020]

[10] 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.

[11] E. Duffield and D. Diaz, "Dash: A Privacy Centric Crypto-Currency", white paper, Aug. pp. 1-16, 2018. http://zioncoins.co.uk/wp-content/uploads/2015/06/Dash-Whitepaper.pdf [accessed: Apr. 20. 2020]

[12] N. V. Saberhagen, "Cryptonote v2.0", white paper, pp. 1-20, Oct. 2013. https://cryptonote.org/whitepaper.pdf. [accessed: Apr. 20. 2020]

[13] RandomX, https://github.com/tevador/RandomX/blob/master/doc/specs.md. [accessed: Apr. 20. 2020]

[14] R. G. Gallager, "Low Density Parity Check Codes", IRE Transactions on Information Theory, Vol. 8, No. 1, pp. 21-28, Jan. 1962.
[15] Bitboin_ECC LDPC, https://github.com/cryptoecc/bitcoin_ECC/blob/ecc-0.1/src/ldpc/LDPC.cpp. [accessed: Apr. 20. 2020]
[16] Bitcoin_ECCChanparams.cpp, https://github.com/cryptoecc/bitcoin_ECC/blob/ecc-0.1/src/ldpc/chainparams.cpp. [accessed: Apr. 20. 2020]

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 블록체인 인터넷경제연구센터 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