Journal of KIIT. Vol. 19, No. 3, pp. 61-70, Mar. 31, 2020. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2021.19.3.61

이더리움 환경을 위한 ECCPoW 블록체인 구현 방법

정현준*, 이흥노**

ECCPoW Blockchain Implementation Method for Ethereum Environment

Hyunjun Jung*, Heung-No Lee**

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-00958) and This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean government (MSIP) (NRF-2021R1A2B5B03002118)

요 약

블록체인 환경에서 트릴레마 문제로 불리는 분산성●보안성●확장성을 동시에 만족시키는 것은 어려운 문제이다. 블록체인들은 트릴레마 문제를 위해 많은 합의 알고리즘를 제안한다. 보안성이 높은 작업증명(Proof-of-Work)은 ASIC 채굴기의 등장으로 분산성이 훼손되었다. 몇몇 블록체인은 ASIC 저항 알고리즘을 개발하였다. 우리는 ASIC 채굴기의 등장을 억제하기 위하여 LDPC 디코더와 해시 함수를 결합한 오류-정정 부호 기반의 작업증명(Error-Correction Codes Proof-of-Work, ECCPoW)를 제안하였다. 그리고 ECCPoW를 구현하는 방법을 제안하였다. 이 논문은 ECCPoW의 코어 1.0 버전에 대하여 설명하고 이더리움 환경에 적용하는 방법을 제안한다. 그리고 우리는 ECCPoW가 적용된 이더리움과 기존의 이더리움(Ethash)의 확장성을 평가한다.

Abstract

The blockchain trilemma problem (Decentralization, Security, Scalability) is difficult to satisfy at the same time in the blockchain environment. Blockchains have proposed many consensus algorithms for the trilemma problem. Proof-of-Work has been damaged by the advent of ASIC miners. Several blockchains have developed ASIC resistance algorithms. We proposed an Error-Correction Codes Proof-of-Work (ECCPoW) that combines the LDPC decoder with a hash function to suppress the appearance of ASIC miner. And we have proposed the implementation of the ECCPoW. This paper describes the core 1.0 version of ECCPoW and proposes a method of application to the Ethereum environment. We evaluate the scalability of Ethereum with ECCPoW and existing Ethereum(Ethash).

Keywords

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

* 군산대학교 소프트웨어융합공학과 교수
- ORCID: https://orcid.org/0000-0002-6717-1395
** 광주과학기술원 전기전자컴퓨터공학부 교수(교신저자)
- ORCID: https://orcid.org/0000-0001-8528-5778

· Received: Sep. 03, 2020, Revised: Mar. 08, 2021, Accepted: Mar. 11, 2021
· 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

이더리움의 창시자 비탈릭 부테린(Vitalik Buterin)은 하나의 블록체인 환경 속에서 분산성•보안성•확장성의 3요소들을 동시에 만족시키는 것은 매우 어려운 문제로 언급하였다. 이를 블록체인의 트릴레마 문제로 정의하였다. 실제로 블록체인 구현에 있어서 트릴레마의 요소 중에서 두 요소를 동시에 만족시키기도 어렵다. 현재 진행하고 있는 암호화폐 프로젝트는 분산성과 보안성 그리고 확장성의 정도를 선택하여 블록체인을 개발하게 된다. 비트코인은 작업증명(PoW, Proof-of-Work)을 사용하여 블록체인의 보안성을 보장한다[1]. 작업증명은 시빌 공격(Sybil attack)과 이중 지불 공격(Double spending attack) 등에 매우 강인하여 금융응용에 적합하다. 작업증명은 금융응용 분야 중 거래 금액이 많고 거래가 자주 발생하지 않는 도메인일수록 적합하다. 작업증명이 분산성까지 갖추게 되면 사토시 나카모토(Satoshi Nakamoto)가 의도한 대로 one CPU one vote에 가깝게 된다. 그러나 채굴기의 등장으로 one ASIC chip은 수천 개의 vote를 담당하게 되었다. 채굴기를 많이 확보한 블록체인은 분산성 실패의 원인이 된다. 분산성 실패는 곧 보안성 약화로 이어지게 된다. 다시 말해, 소수의 몇 사람이 좌지우지하는 블록체인으로 전락하고, 이때 이중 지불 공격하기가 쉬워진다. 이런 상황에서 초당 거래속도(TPS, Transactions Per Second) 및 블록 크기를 확장하려는 시도는 실패할 수밖에 없다. 비트코인의 라이트닝 네트워크[2], 이더리움의 샤딩[3], 사이드체인 등의 확장성 향상을 위한 방법들은 현실적인 한계를 갖게 된다. 하나의 블록체인 환경 속에서 3요소를 모두 만족시키려는 것은 한계에 부딪히게 된다.

기존의 블록체인 합의 알고리즘을 살펴보면, 작업증명계열이 높은 보안성을 확보 가능하다[4][5]. 그러나 비트코인과 같은 성공한 블록체인의 경우에 ASIC/FPGA 채굴기의 등장으로 작업증명의 해시 레이트가 지속해서 높아지게 된다. 채굴기를 다량으로 확보한 채굴기업과 집단의 등장으로 탈중앙화마저 위협받게 되었다. 시간이 흘러 작업증명 알고리즘은 두 가지 문제가 대두되었다. 첫째, 채굴에 천문학적 수준의 전기 에너지를 소모한다. 둘째, 일반 개인의 채굴 참여가 어렵게 되어 탈중앙화에 실패 위험이 발생한다. 즉, 대규모 ASIC 채굴기를 다수 확보한 소수의 채굴업자에 의해 변경될 위협이 발생한다.

ASIC 채굴기로 인하여 블록체인에 생기는 문제는 다음과 같다. 분산성 위협으로써, 거래 내역의 블록체인 입력과 기록 보존을 소수의 채굴업자가 수행하게 되므로 탈중앙화에 실패하게 된다. 무결성 위협으로 소수의 채굴업자가 지배하는(해시파워 51% 이상) 블록체인은 그중 몇 사람만 의기투합할 때 거래 수수료 인상, 거래기록의 선택적 수납, 거래기록의 삭제 등도 가능하게 된다. 블록체인에 기록되는 데이터는 영속적이며 기록되었던 순전 무결하게 보존된다는 무결성 신뢰에 심각한 위협이 된다.

우리는 블록체인의 보안성과 분산성을 동시에 확보하기 위하여 ECCPoW(Error-Correction Codes PoW) 기반의 블록체인을 제안하였다[6]. 그리고 ECCPoW 블록체인의 구현방법에 대하여 제안하였다[7]. 그리고 제안한 ECCPoW를 비트코인에 구동하는 방법에 대하여 제안하였다[8].

이 논문은 ECCPoW 코어 1.0 버전에 대하여 설명한다. 이를 통해, ECCPoW에서 제안하는 구성 요소들의 역할에 대하여 이해할 수 있다. 그리고 ECCPoW를 이더리움에 적용하는 방법을 제안한다. 이전 논문에서 우리는 비트코인에 적용하는 방법에 대하여 다루었다. 이더리움은 비트코인에서는 지원하지 못하는 스마트 컨트랙트(Smart contract)를 지원한다. 그리고 ECCPoW가 적용된 이더리움과 기존의 이더리움(Ethash)을 확장성 측면에서 평가한다.

이 논문은 다음과 같이 구성되어 있다. 2장에서는 ASIC 저항성과 관련된 연구를 소개한다. 3장은 ECCPoW 코어 1.0 버전에 대한 설명과 ECCPoW를 이더리움(ECCETC)에 적용하는 방법에 대하여 제안한다. 4장에서는 ECCETC를 이더리움(Ethash)과 확장성 측면에서 평가한다. 5장에서는 결론을 제시한다.

Ⅱ. 관련 연구

이 장에서는 ASIC 저항성에 관련된 연구를 요약한다. 몇몇 블록체인들은 ASIC 저항성을 고려하여 설계되고 지속해서 하드포크를 통하여 관리한다.

ASIC 저항성을 고려하는 블록체인의 특징과 방향성에 대하여 다루고자 한다.

이더리움은 비탈릭 부테린이 2015년 개발한 분산 컴퓨팅 플랫폼이다[9][10]. 2009년 비트코인의 등장으로 블록체인에 관심이 쏠렸다. 비트코인은 사용자가 이용할 수 있는 자유 공간인 OP_RETURN을 제공하였다. 하지만 블록체인을 이용한 기능을 구현하는 데에 한계점을 느끼고 이더리움을 개발하였다. 이더리움은 트랜잭션에 기존의 거래내용과 스마트 컨트랙트의 프로그램 코드가 저장된다. 스마트 컨트랙트는 특정 조건이 충족되면 실행되는 계약을 블록체인상에 구현할 수 있다. 이더리움은 이더해시(Ethash) 알고리즘을 사용한다. 이더리움은 ASIC 저항성을 위하여 Ethash 알고리즘은 비선형 그래프(DAG, Directed Acyclic Graph)를 이용한다. Ethash는 일정한 크기의 메모리(초창기 1GB에서 현재 3.99GB)를 블록을 생성할 시 참조하도록 설계되었다. 이를 통하여 ASIC에 비효율적으로 동작하도록 유도하였다. 블록의 유효성은 통과되었지만, 최종 블록으로 인정받지 못한 고아 블록(엉클 블록, uncle block)에도 보상을 준다. 엉클 블록에 보상을 줌으로써 직접적인 채굴 참여를 유도하여 안정적인 네트워크를 운영한다.

이더리움에 대응하는 ASIC이 등장하자 분산성을 확보하기 위하여 Ethash 알고리즘 대신 ProgPoW[11]의 적용을 승인하였다. ProgPoW는 두 가지 특징을 가지고 있다. 첫째, 이더리움의 분산성을 위하여 채굴에 사용되는 문제를 정기적으로 변경한다. 두 번째, 채굴 알고리즘이 그래픽 카드의 모든 구성 요소를 최대한 활용하도록 설계하였다. ProgPoW는 블록 번호를 기준으로 임의로 문제를 생성하고, 생성된 문제를 ASIC 채굴보다 GPU에 효율적으로 동작한다. 그리고 정기적으로 GPU가 빠르게 적응할 수 있는 문제로 변경하면서 ASIC에 대한 성능 차이를 확보한다.

이더리움은 1.0에서 이더리움 2.0으로 전환을 시도하고 있으며, 사용되는 알고리즘을 작업증명(PoW)에서 지분증명(PoS)으로 업그레이드를 계획하고 있다. 이더리움은 급격한 지분증명의 변화에 따른 부작용에 대비하기 위하여 변환을 단계적으로 진행할 예정이다. Casper the FFG(Friendly Finality Gadget)은 블록 제안 메커니즘위에 구현되며 캐스퍼의 첫 번째 버전은 작업증명 방식 위에 지분증명을 구현하는 Hybrid PoW/PoS 방식으로 구현한다[13]. PoS는 규칙을 어기는 검증자를 식별하고 이들의 예치금을 몰수(Slashing)하는 방식으로 책임을 물을 수 있으며 이를 통해 nothing-at-stake 문제를 해결한다. 그리고 검증자의 집합이 동적으로 변경된다.

샤딩(Sharding)은 확장성 문제 솔루션 중 하나로 이더리움이 PoS 합의 알고리즘으로 전환할 것을 기반으로 설계 되었다[14]. 샤딩은 플라즈마(Plasma), 트루빗(Trubit), 라이덴 네트워크(Raiden Network), 라이트닝(Lightning network) 등과 마찬가지로 확장성 문제를 해결하기 위해 제안된 솔루션이다. 플라즈마, 라이덴 네트워크는 Off-chain 솔루션(Layer-2솔루션)인 반면 샤딩은 On-chain 솔루션(Layer-1 솔루션)이다. 샤딩은 전체 네트워크를 분할한 뒤 트랜잭션을 영역별로 저장하고 이를 병렬적으로 처리하여 블록체인에 확장성을 부여하는 솔루션으로 데이터를 샤드라는 단위로 나눠서 저장 및 처리한다. 이더리움에서 샤딩은 메인체인을 k개의 샤드로 분할하며 각 샤드는 네트워크상의 전체 트랜잭션을 분류하여 병렬적으로 처리한다. 이는 기존에 하나의 메인체인이 모든 트랜잭션을 순차적으로 처리하던 것과 비교되며 이 방식으로 네트워크의 전체 처리량은 샤드의 배수만큼 향상된다.

모네로(Monero)는 PoW 알고리즘으로 크립토나이트(CryptoNight)을 사용하였다. 모네로는 거래의 익명성을 위하여 CryptoNote 프로토콜을 채택하였다[12]. CryptoNote의 작동원리는 거래가 시작되면 특정 그룹 내에서 키가 섞이는 링 시그니처(Ring signature)기술을 사용한다. 링 시그니처를 통과한 거래내역들은 누가 누구에게 얼마를 보냈는지 알 수 없다. 거래내역을 조회하려면 private key를 통해서 확인할 수 있다. 모네로는 ASIC 채굴기에 대한 저항성을 확보하기 위하여 6개월마다 한 번씩 하드포크를 진행하였다. 하지만 잦은 하드포크는 채굴자의 이탈을 일으키었고 오히려 채굴의 집중화를 가져올 위기가 높아졌다. RandomX는 키-블록개념을 이용하여 기존의 정기적인 하드포크 없이 정기적으로 채

굴 방법에 변화를 준다[15].

레이븐(Ravencoin)은 ASIC 채굴을 피하기 위하여 X16R 알고리즘을 사용한다[16]. X16R은 X-11 계열의 암호화폐 채굴 알고리즘이다[17]. X11은 ASIC을 억제하기 위해 다수의 해시 함수를 사용하여 심층과 복잡성을 추가하였다. X16R은 16개의 알고리즘을 랜덤하게 사용한다. KawPoW는 ProgPoW와 비슷하며 Raven에 대한 특수한 매개변수가 적용되었다. 레이븐은 GPU 채굴자에게 전력을 돌려주는 알고리즘을 제공한다. 그래서 2GB의 RAM이 있는 구형 GPU 모델을 사용하여 채굴할 수 있다. 레이븐은 수백 개의 GPU와 제로 ASIC를 목표로 한다.

III. ECCPoW 코어 1.0 버전 및 이더리움 적용방법

3.1 ECCPoW 코어 1.0 버전 소개

ECCPoW 코어 1.0 버전은 오류 정정부호(Error correction codes) 중의 하나인 저밀도 패리티 체크(LDPC, Low Density Parity Check)를 사용하여 매 블록 무작위로 변하는 암호 퍼즐을 생성하는 방법을 포함한다. 그리고 퍼즐을 푸는 디코더와 퍼즐의 난이도 및 증감 방법에 관한 연구를 포함한다. 이장에서는 기존에 제안한 ECCPoW의 동작 방법을 요약하여 보여준다[6].

그림 1은 ECCPoW가 적용된 블록체인의 한 예를 보여준다. ECCPoW 코어 1.0 버전은 세 가지 주요 구성 요소를 포함한다.

첫 번째 구성 요소는 암호 퍼즐 생성 부분이다. ECCPoW의 암호 퍼즐은 LDPC 부호를 이용해 생성되고, 부호는 LDPC 패리티 체크 행렬(Parity check matrix)에 의해 정의된다. ECCPoW에서 패리티 체크 행렬 변경은 부호의 변경이고, 곧 암호 퍼즐 변경을 의미한다. ECCPoW는 채굴될 블록에 포함된 이전 해시값을 이용해 매 블록 패리티 체크 행렬을 무작위로 생성하는 방법을 설계하였다.

두 번째 구성 요소는 암호 퍼즐 디코더 부분이다. ECCPoW가 만든 암호 퍼즐을 풀기 위한 요소이다. 디코더는 해시의 출력값과 첫 번째 구성 요소인 암호 퍼즐 생성의 결과에 따라 생성된 패리티 체크 행렬을 입력값으로 사용한다. 그리고 메시지 전달 알고리즘에 기초한 디코딩을 수행한 후 결과를 출력한다. ECCPoW 블록체인은 출력값을 기준으로 암호 퍼즐의 해결 여부를 판단한다.

세 번째 구성 요소는 암호 퍼즐 난이도 조절 부분이다. ECCPoW는 암호 퍼즐의 난이도를 패리티 체크 행렬의 변수들을 이용해 확률값으로 제시한다. 확률분석을 토대로 암호 퍼즐 난이도 증감 방법을 제시하였다.

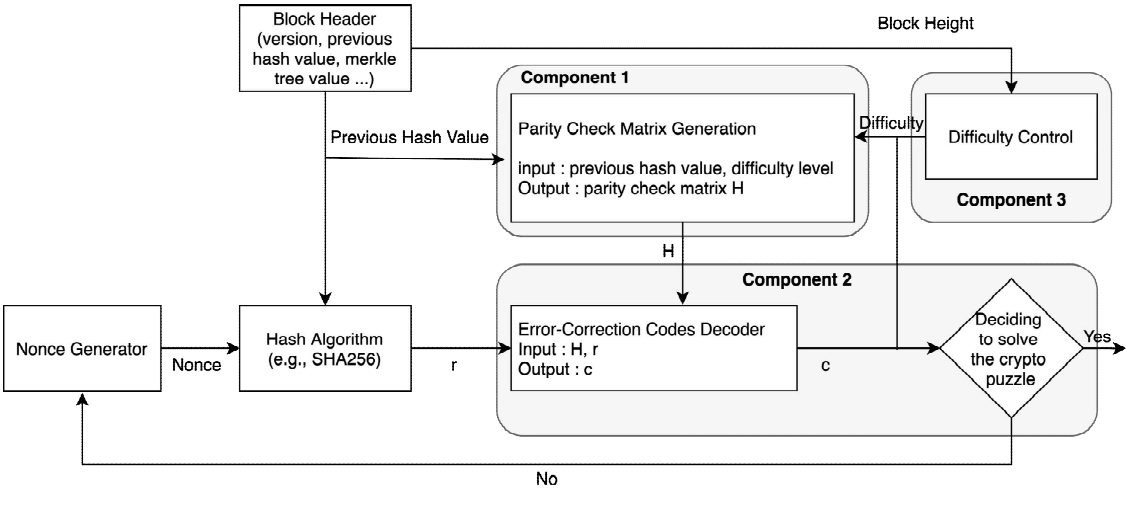

그림 1. ECCPoW가 적용된 블록체인 예
Fig. 1. Example of a blockchain with ECCPoW

ECCPoW는 분산성과 보안성을 확보하기 위하여 제안되었다. 대부분의 작업증명 기반의 블록체인에서는 채굴에 특화된 ASIC 채굴기가 사용되고 있다. 대량의 ASIC 채굴기를 확보한 소수의 채굴업자들이 채굴 시장을 장악함으로써 분산성이 훼손되었다. 소수의 채굴업자들이 담합을 하면 이중 지불 및 블록 리오그(Reorg) 공격 등이 가능해지므로 블록체인의 보안성이 훼손된다. ECCPoW는 분산성과 보안성 훼손의 근본적 원인인 ASIC 사용을 막기 위해 개발된 작업증명 기술이다. 이 기술이 적용된 블록체인에서는 CPU/GPU 사용만 허용되므로 채굴 참여자의 숫자를 극대화할 수 있다. 이를 통해 분산성 확보가 가능하다. 또한, 채굴업자들에 의한 담합을 방지하여 보안성을 확보하다. ECCPoW 코어 1.0 버전에서는 암호 퍼즐의 생성과 그것을 푸는 디코더와 암호 퍼즐의 난이도를 조절할 수 있는 기능이 구현되었다.

ECCPoW는 해시 함수와 부호 디코더의 합성 함수로 구성되어 있다. ECCPoW는 오류 정정 부호와 작업증명을 동시에 사용하여 분산성과 보안성을 확보하기 위해 개발된 작업증명 기술이다. 해시 함수(예, SHA256)와 오류 정정성 함수를 사용한다.

그림 2는 ECCPoW가 사용하는 합성 함수를 보여준다. 해시 함수는 블록 헤더를 입력값으로 받아 해시값 r을 출력한다. 오류 정정 부호 디코더는 해시 r과 디코딩 과정에서 사용되는 LDPC 패리티 체크 행렬 H를 입력값으로 받는다. 디코더는 정해진 규칙에 따른 디코딩 알고리즘을 실행하고 결과 값인 c를 출력한다.

ECCPoW는 작업증명 알고리즘으로 탈중앙성과 보안성 확보에 도움을 준다. ECCPoW의 시변성과 무한성으로 인해 암호 퍼즐을 만들 때 사용되는 합성 함수를 매 블록 변화시킬 수 있다. 합성 함수의 변경은 부호 디코더 부분의 패리티 체크 행렬의 크기 등을 변경함으로써 이루어진다. 다수의 패리티 체크 행렬들을 지원하는 ASIC 디코더는 추가적인 메모리, 스위치 장치 등의 별도의 하드웨어가 필요하고, 디코더의 크기 및 제작비용을 증가시키는 요인이라고 발표되었다[18]. 즉, 시변성과 무한성 때문에 부호 디코더를 ASIC로 개발하는 것은 불가능하다. 따라서 ASIC 채굴기 개발 억제가 가능해지고 결국 참여자의 수를 극대화함으로써 탈중앙성 확보가 가능하다. 또한 ASIC 채굴기 개발 억제로 인한 블록체인을 변경할 수 있는 소수의 채굴기업 등장을 억제할 수 있다. 이는 이들이 유발할 수 있는 담합에 의한 이중 지불 공격, 블록 리오그 공격 등을 예방할 수 있다. 이는 블록체인의 보안성을 유지해준다.

3.2 ECCPoW 코어1.0의 이더리움 시스템 적용

이 장은 제안한 ECCPoW 코어 1.0 버전을 이더리움 환경에 적용방법에 대하여 보여준다. 이더리움은 스마트 컨트랙트를 지원하는 대표적인 2세대 블록체인이다. ECCPoW는 작업증명을 사용하는 블록체인에 적용할 수 있다. 이더리움은 Ethash 작업증명 알고리즘을 사용한다. 이더리움은 ASIC에 의한 분산성이 훼손되자 Ethash 알고리즘 대신 ProgPoW를 적용하기로 하였다. 우리는 이 장을 통하여 ECCPoW를 기존 블록체인에 적용하는 방법의 방향성을 제시하고자 한다.

이더리움은 ASIC를 방지하기 위하여 채굴 알고리즘(Ethash)과 채굴 보상(엉클 블록 보상)을 가지고 있다. 하지만 이더리움의 Ethash를 지원하는 Antminer E3(180Mh) 채굴기가 2018년에 등장하였다[19].

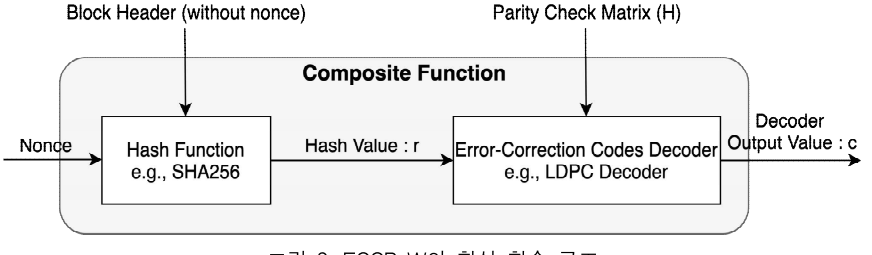

그림 2. ECCPoW의 합성 함수 구조
Fig. 2. Composite function structure of ECCPoW

E3는 NVIDIA의 GPU 중에서 고급라인인 GTX TITIAN 9개를 사용했을 때와 동일한 성능이다[20]. 이더리움에서 ASIC이 등장하자 블록생성을 위하여 요구하는 컴퓨팅 파워는 점점 높아지고 보상은 적어지게 되었다. Ethash는 ASIC 채굴에 저항성을 갖고 GPU로 공정하게 채굴할 수 있도록 고안되었다. 하지만 ASIC 채굴기를 사용하는 사용자가 증가한다면 상대적으로 블록채굴이 낮은 GPU 사용자들은 점차 블록생성을 포기할 것이다. 이는 이더리움 블록체인의 채굴 중앙화를 가져온다.

ECCPoW는 ASIC채굴기의 등장을 해결하기 위해 LDPC 디코더와 해시함수를 결합한 오류 정정 부호 기반의 작업증명이다. 일반적으로 채굴 알고리즘은 블록마다 해시함수(SHA256, Keccak 등)를 수행하여 블록을 생성한다. ASIC 채굴기는 해시함수를 최적으로 수행할 수 있도록 설계한다. ECC PoW는 매 블록 다른 문제를 푸는 방법을 제안하였다. 우리는 ECCPoW를 비트코인에 적용하여 보안성과 확장성을 유지하며 채굴 성공률을 52% 높였다[6].

ECCPoW를 이더리움에 적용할 경우 매 블록 다른 해시함수를 푸는 효과를 기대할 수 있다. ECCPoW는 이전 블록의 내용 중 일부를 seed 값으로 사용하여 블록생성의 문제로 사용한다. 이더리움은 ECCPoW를 적용한다면 강력한 ASIC 저항성을 확보할 수 있습니다. 최근 ASIC 저항성 방법은 블록체인의 채굴 알고리즘을 주기적으로 변형하고자 한다. 이더리움의 주기적 변경이 가능한 ProgPoW의 개념보다 더 강력한 ASIC 저항성을 제공할 수 있다.

이더리움에서 ECCPoW는 블록이 생성될 때 영향을 준다. 기존의 go-ethereum의 Consensus 패키지 내부에 ECCPoW 패키지를 추가한다. 그림 3은 이더리움 시스템에 적용된 ECCPoW 코어 패키지 구성도이다. 이더리움의 합의 계층 내부 구조를 변경하기 위하여 ECCPoW 패키지는 algorithm. go, algorithm_ test.go, api.go, consensus.go, consensus_test.go, ethash.go, ethash_test.go, sealer.go, sealer_test. go, LDPCDecoder. go, LDPCDecoder_test.go, LDPC_ utils. go 등 12가지로 구성되어 있으며, 각각의 PoW합의 계층을 구성하는 세부적용 요소이다.

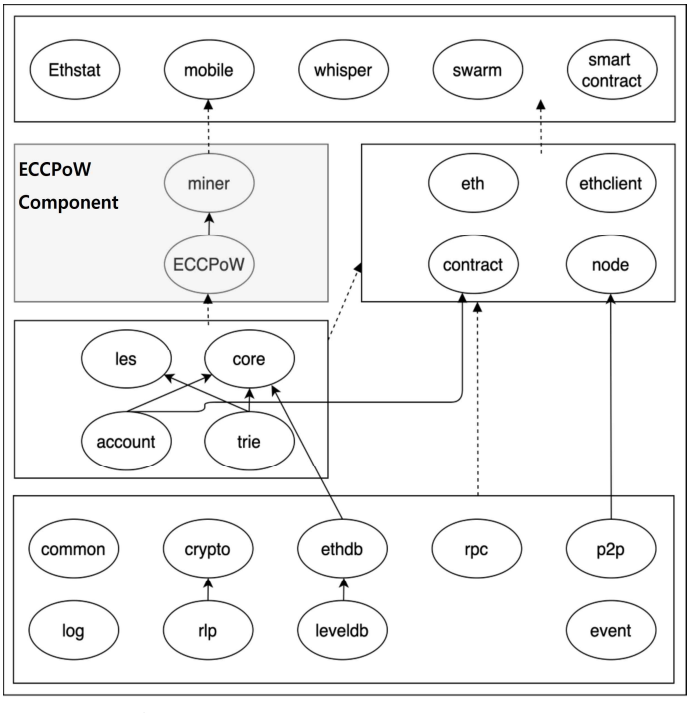

그림 3. 이더리움 환경에 적용된 ECCPoW 코어 패키지 구성도
Fig. 3. ECCPoW core package diagram for Ethereum environment

Algorith.go, Algorithm_test.go : 합의 알고리즘 구현의 핵심 패키지로 캐시 생성 및 크기 설정, 해시 연산 대상 데이터 세트 크기와 형식 지정, 해시 산출, 시드 해시 계산, 핵심 알고리즘 등의 기능으로 구성되어 있다. 특히 마이닝 과정의 핵심 함수이며 논스를 찾는 과정을 담당하는 hashimoto 함수가 이 패키지에 포함되어 있다. hashimoto 함수에서 ECCPoW 디코딩을 수행한다.

Api.go : 알고리즘에 의해서 만들어진 타겟 값 논스, 다이제스트, 해시, 에러)과 현재의 해시 레이트, 타겟 해시 레이트 등을 메시지화 하여 패키지 외부로 전달하는 역할을 수행한다.

Consensus.go, Consensus_test.go : 블록보상, 블록타임, 엉클카운터, 난이도(difficulty) 및 각종 블록의 포크 프로토콜을 담아 타겟 블록을 계산 및 검증하는 역할을 수행한다. Timestamp, uncles, ancestor block, difficulty rate, 다이제스트 값, proof of work등이 바르게 계산되었는지를 확인한다.

LDPCDecoder.go, LDPCDecoder_test.go, LDPD_utils.go : C++로 작성된 ECCPoW를 golang으로 포팅한 부분이다. PCM 생성, 해시 벡터 생성, LDPC 디코딩 등 ECCPoW의 핵심적인 연산을 수행한다.

Sealer.go, sealer_test.go : 저장된 컨센서스 엔진, 블록타입, 체인을 파라미터로 받아 이를 감싸 새로운 블록을 만들고, 앞으로 만들어질 블록에 대한 타겟해시, 논스 탐색 등의 기능을 가진 패키지이다.

Cmd/puppeth : 이더리움 프라이빗 네트워크 매니저인 Puppeth를 수정하여 합의 알고리즘을 ECCPoW를 사용하는 프라이빗 네트워크를 손쉽게 배포할 수 있도록 한다.

그림 4는 이더리움에서 ECCPoW가 동작하는 프로세스를 보여준다. 이더리움은 비트코인의 SHA256 같은 Keccak-256(SHA-3의 변형된 형태) 해시 함수를 사용한다. 우리는 Nonce를 Keccak 함수의 입력으로 Seed 값을 생성한다. 그리고 Seed 값을 이용하여 해시 벡터를 생성한다. 마이닝을 위하여 ECCPoW에서 필요한 패리티 체크 행렬과 Nonce를 디코더에 입력한다. 마지막으로 출력 값이 Codeword인지를 판단하여 Codeword 이면 마이닝을 종료하고, Codeword가 아닐 시 Nonce를 다시 생성하여 프로세스를 반복한다.

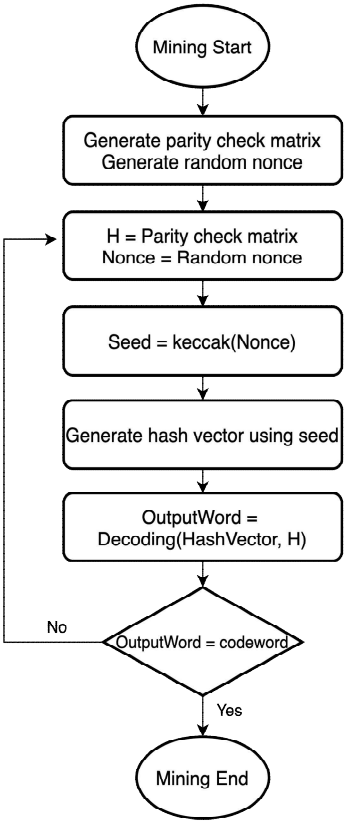

그림 4. 이더리움 환경에서의 ECCPoW 프로세스
Fig. 4. ECCPoW process in Ethereum environment

IV. 평 가

이 장은 ECCPoW가 적용된 이더리움과 기존의 이더리움을 확장성 측면에서 비교한다. TPS는 블록체인에서 확장성을 측정하는 요소이다.

우리는 ECCPoW를 비트코인에 적용하는 연구를 진행하였다[8]. 이 연구를 통하여 우리는 ECCPoW를 적용했을 때 보안성을 유지한 상황에서 분산성을 확보할 수 있다는 것을 확인할 수 있었다. 이 장에서는 이더리움의 Ethash 알고리즘 대신 ECCPoW를 적용하였을 때에 확장성에 영향을 주는지를 확인하고자 한다.

평가 환경은 Ubuntu Server 16.04 LTS, m5.xlarge (vCPU processor 4 core, RAM 16GB), SSD 8GB에서 진행하였다. ECCPoW를 적용한 이더리움에 사용할 인스턴스 4대와 이더리움 합의 알고리즘을 사용하는 인스턴스 4대를 준비하여 실험하였다. 그림 5는

평가에 사용한 AWS(Amazon Web Services) 인스턴스 리스트이다.

이더리움의 기존 합의 알고리즘인 Ethash와 ECCPoW의 직접적인 비교를 위하여 8개의 인스턴스를 생성하여 각 4대씩 사용하여 Ethash를 사용하는 이더리움 네트워크와 ECCPoW를 사용하는 이더리움 네트워크를 구성한다. 각 네트워크가 사용하는 인스턴스에는 공통으로 블록생성을 담당하는 마이닝 노드를 설치하였다. 네트워크 별로 한 대의 노드에만 노드 간 연결을 담당하는 bootnode와 네트워크 모니터링을 담당하는 ethstat을 설치하였다. 확장성 테스트를 위해 트랜잭션을 발생시키는 오픈소스인 chainload를 사용하였다. chainload를 이용해 각 마이닝 노드에 같은 빈도의 트랜잭션을 전송하였다.

우리는 같은 ECCPoW가 적용된 이더리움과 Ethash가 적용된 이더리움을 같은 환경에서 테스트하였다. 그리고 테스트 결과 중 일부의(1시간) 트랜잭션 처리량을 비교하였다. 각 테스트가 시작하는 시점의 블록 번호와 끝나는 시점의 블록 번호를 기록하고 해당 시간 동안 트랜잭션 처리량을 비교하는 방식으로 진행하였다. TPS는 블록당 트랜잭션의 개수를 a라 했을 때 다음과 같이 정의한다. n은 테스트 횟수이다.

$$TPS = \frac{a_1 + a_2 + a_3 + \ldots + a_n}{3600} \quad (1)$$

표 1은 Ethash와 ECCPoW 블록체인의 TPS 측정 결과를 보여준다. 같은 요건을 가진 AWS 인스턴스에 같은 숫자의 이더리움 블록체인(Ethash)과 ECCPoW 블록체인 노드를 구축하였다. 그리고 동일한 양의 트랜잭션을 전송하여 두 네트워크의 트랜잭션 처리량을 비교하였다. 실험결과 ECCPoW는 22.325 TPS, Ethash는 21.681 TPS를 기록하였다. ECCPoW는 Ethash 대비 확장성 102.97%의 성능을 보였다. ECCPoW는 이더리움에서 Ethash를 대신했을 때 확장성 측면에서 비슷한 성능을 보여주는 것을 알 수 있다. 테스트 시간을 증가하여도 비슷한 성능을 유지하였다.

표 1. Ethash와 ECCPoW 블록체인 TPS 비교
Table 1. TPS comparison of Ethash and ECCPoW blockchain

Number	Blockchain	Transaction volume in block	Block generation time	TPS
1	Ethash	13,489	600 s	22.481
	ECCPoW	13,229	600 s	22.165
2	Ethash	13,312	600 s	22.186
	ECCPoW	13,168	600 s	21.946
3	Ethash	12,652	600 s	21.086
	ECCPoW	13,764	600 s	22.940
4	Ethash	11,084	600 s	18.473
	ECCPoW	11,051	600 s	18.418
5	Ethash	13,732	600 s	22.886
	ECCPoW	15,776	600 s	26.293
6	Ethash	13,783	600 s	22.971
	ECCPoW	13,314	600 s	22.190
Average	Ethash	13,008.6	600 s	21.681
	ECCPoW	13,396.3	600 s	22.325

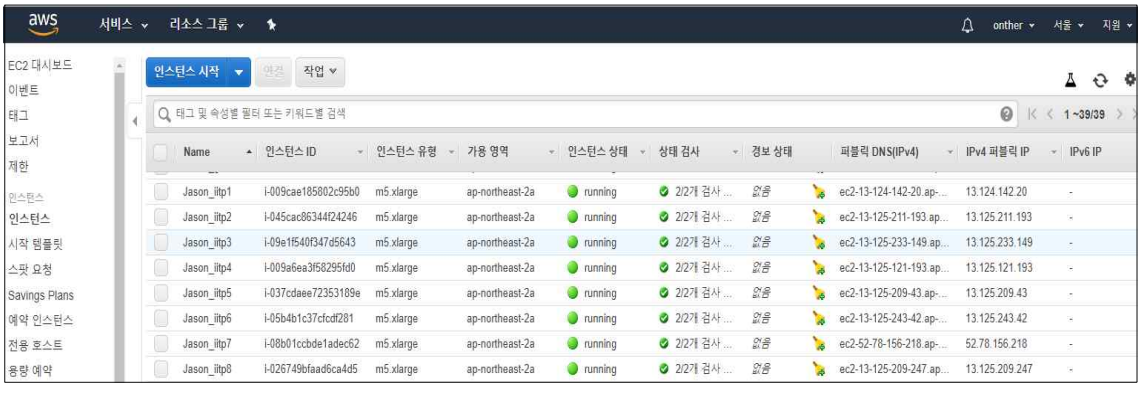

그림 5. Amazon Web Services 인스턴스 리스트
Fig. 5. List of Amazon Web services instances

우리는 실험에서 보안성은 이더리움과 같은 시스템을 사용하기 때문에 같다고 가정한다. 실험결과는 ECCPoW 알고리즘을 이더리움에 적용할 때 기존의 이더리움의 확장성 측면에서 손해는 없으면서 분산성을 확보할 수 있음을 보여준다.

ECCPoW를 이더리움의 Ethash를 대신하여 구현하고 확장성을 평가하였다. 이더리움은 현재의 PoW를 사용하는 1.0에서 PoS를 사용하는 이더리움 2.0으로 변화한다. ECCPoW는 PoW에서 사용할 수 있는 매 블록 다른 문제를 푸는 효과를 가져온다. 이더리움은 확장성의 한계를 극복하기 위하여 On-chain 솔루션(Sharding 등), Off-chain 솔루션(Plasma, Trubit, Verification Game 등)을 제한하였다. 이 방법들은 PoW에는 적용할 수 없으며 PoS가 적용한 후 도입 여부를 판단할 수 있다. ECCPoW는 PoW가 사용되거나 PoS와 공동으로 사용될 때의 대안이 될 수 있다. 이더리움의 확장성(블록생성 15~20초) 사이에서 공정한 블록생성을 도와줄 수 있을 것이다.

V. 결 론

블록체인 환경에서 트릴레마 요소인 분산성, 보안성 그리고 확장성을 동시에 만족시키는 것은 어렵다. 블록체인은 작업증명(Proof-of-Work), 지분증명(Proof-of-Stake), 위임된 지분증명(Delegated Proof-of-Stake) 등 여러 합의 알고리즘이 제안되었다. 보안성이 높은 작업증명 블록체인은 ASIC/FPGA 채굴기의 등장으로 해시 레이트가 지속해서 높아졌다. 그리고 일반 개인의 채굴 참여가 어렵게 되어 분산성이 회손되어 탈중앙화에 문제가 생기게 되었다. 우리는 블록체인의 보안성과 분산성을 동시에 확보하기 위하여 ECCPoW 블록체인을 제안하였다. 이 논문은 제안한 ECCPoW 버전 1.0에 대하여 설명하고 이더리움에 적용하는 방법을 보여주었다. ECCPoW는 세 가지 핵심 구성 요소인 암호 퍼즐 생성, 암호 퍼즐 디코더, 암호 퍼즐 난이도 조절을 이용하여 매 블록 암호 퍼즐을 변경한다. 매 블록 변경되는 성질을 이용하여 ECCPoW를 사용하는 블록체인은 ASIC 채굴기 등장을 억제한다. ECCPoW가 적용된 이더리움과 기존의 이더리움(ethash)을 확장성 측면에서 비교하였다. 비교 결과 ECCPoW는 Ethash와 비교하여 102.97% 높은 성능을 보였다. ECCPoW는 분산성을 높이려는 제안방법을 수행하여도 확장성을 유지하는 것을 확인할 수 있었다. ECCPoW는 기존의 블록체인들이 ASIC 저항성을 확보하기 위하여 선택할 수 있는 하나의 대안이 될 수 있을 것이다.

References

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

[2] J. Poon and T. Dryja, "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments", 1. 2016, https://lightning.network/lightning-network-paper.pdf

[3] 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. https://doi.org/10.1109/ACCESS.2020.2967218

[4] I. G. A. K. Gemeliarana, and R. F. Sari, "Evaluation of Proof of Work (POW) Blockchains Security Network on Selfish Mining", in 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Nov. 2018, pp. 126-130, https://doi.org/10.1109/ISRITI.2018.8864381.

[5] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the Security and Performance of Proof of Work Blockchains", in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, pp. 3-16, Oct. 2016. https://doi.org/10.1145/2976749.2978341.

[6] 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.

[7] S. Park, H. Choi, and H. N. Lee, "Time-Variant Proof-of-Work using Error-Correction Codes", Submitted to IEEE Trans. on Information Forencis and Security. (https://arxiv.org/abs/2006.12306)

[8] H. Jung and H. N. Lee, "ECCPoW: Error-Correction Code based Proof-of-Work for ASIC Resistance", Symmetry, Vol. 12, No. 6, Art. no. 6, Jun. 2020, https://doi.org/10.3390/sym12060988.

[9] 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]

[10] 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]

[11] IfDefElse and G. Colvin, "ProgPoW, a Programmatic Proof-of-Work", Ethereum - EIPs, No. 1057, May 2018.

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

[13] V. Buterin and V. Griffith, "Casper the Friendly Finality Gadget", Jan. 2019, https://arxiv.org/abs/1710.09437

[14] Sharding Introduction R&D Compendium, https://eth.wiki/en/sharding/sharding-introduction-r-d-compendium

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

[16] B. Fenton and T. Black, "Ravencoin: A Peer to Peer Electronic System for the Creation and Transfer of Assets", April. 2018, https://ravencoin.org/assets/documents/Ravencoin.pdf. [accessed: Apr. 20. 2020]

[17] T. Black and J. Weight, "X16R ASIC Resistant by Design", https://ravencoin.org/assets/documents/X16R-Whitepaper.pdf. [accessed: Apr. 20. 2020]

[18] S. Shao, P. Hailes, T. Y. Wang, J. Y. Wu, R. G. Maunder, B. M Al-Hashimi, and L. Hanzo, "Survey of Turbo, LDPC and Polar De-coder ASIC Implementation", IEEE Communications Surveys & Tutorials 2019

[19] Antminer E3, https://en.wikipedia.org/wiki/Bitmain, [accessed: 30. Mar. 30. 2021]

[20] Ethereum mining calculator, https://etherscan.io/ether-mining-calculator. [accessed: Mar. 30. 2020]

저자소개

정 현 준 (Hyunjun Jung)

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

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