이더리움(Ethereum) 소개 및 특징 분석

(보안연구부 보안기술팀 / 2016.03.04)

□ 개 요

o 블록체인(Blockchain)에 대한 관심이 높아짐에 따라 다양한 산업에서 이를 활용하기 위해 연구와 투자가 이루어지고 있으며, 특히 이더리움은 비트코인(Bitcoin)의 진화한 기술로 주목받기 시작함

< 이더리움과 대표적 연계 사례 >

- 마이크로소프트는 자사 클라우드 플랫폼(Azure) 사용자를 위한 블록체인 서비스(BaaS)를 출시하기 위해 이더리움 코딩그룹(ConsenSys)과 협약(2015.11)
- R3 CEV는 마이크로소프트 클라우드 플랫폼 기반 블록체인 서비스에 이더리움을 이용한 서비스 개발이 진행중이며, 현재는 1차 테스트를 마친 상황(2016.2)

o 본 보고서에서는 비트코인과 다른 이더리움의 새로운 개념, 기술에 대해 소개하고자 함

□ 이더리움 개요

o **(등장배경)** 비트코인이 화폐로서의 신뢰성을 보장하기 위한 체계를 구축하였지만, 시스템을 확장·개선하기 위한 협의가 이루어지지 않아 지속적으로 실패론이 대두됨

- 비트코인을 이용한 서비스 개발에는 한계가 있으며, 추가 기능을 개발하기 위해서는 상당량의 코드 수정이 필요하므로 매우 비효율적

o **(개념)** 2013년 비탈릭 뷰테린(Vitalik Buterin)에 의해 고안되었으며, 프로그래밍이 가능한 블록체인(Programmable Blockchain)을 구현한 웹 프레임워크로써, 확장된 분산어플리케이션을 만들 수 있는 플랫폼을 제공하는 것임

- 또는 튜링완전언어(Turing-Complete Programming Language)를 제공하는 플랫폼이라고도 불림

참고 : 튜링 머신/완전성/완전언어

□ **튜링머신(Turing-Machine)**이란 순서에 따라 계산이나 논리 조작을 행하는 장치로, 적절한 기억 장소와 알고리즘만 주어진다면 어떠한 계산이라도 가능함을 보여 주어 현대 컴퓨터의 원형을 제시

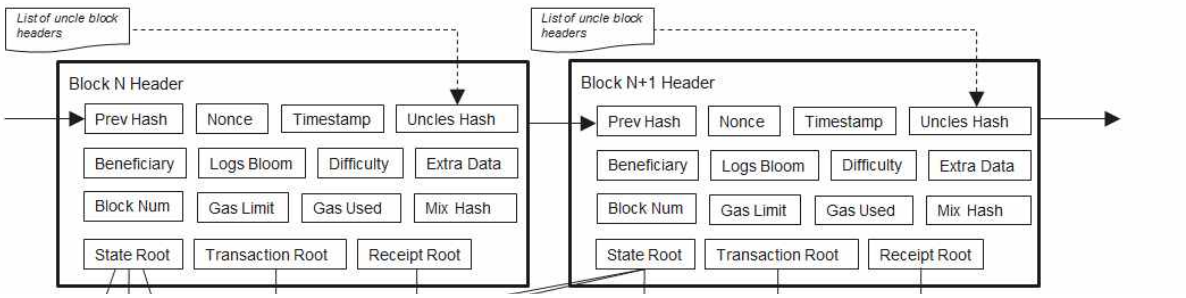
□ **튜링완전성(Turing-Completeness)**이란 어떤 프로그래밍 언어나 추상 기계가 튜링 기계와 동일한 계산 능력을 가진다는 의미로, 튜링머신으로 풀 수 있는 문제, 즉 계산적인 문제를 그 프로그래밍 언어나 추상 기계로 풀 수 있다는 의미

□ **튜링완전언어**란 조건 분기문(if, goto 등)이 존재하고, 루프문(for, while 등)은 모두 조건 분기문으로 바꿔 쓸 수 있으며, 메모리의 임의 위치 값을 변경할 수 있는 언어를 일컬으며, C/C++, Pascal 등이 해당

o **(구조)** 비트코인의 구조와 유사하게 블록의 번호, 이전 블록의 정보, 해시 트리, 트랜잭션 정보, 넌스(Nonce) 등을 포함하고 있으며, 엉클(Uncle)블록*과 연료(Gas, 이더리움의 화폐와 동일) 등의 개념이 추가됨

* 비트코인의 고아(Orphan)블록과 동일한 개념

< 이더리움 블록체인의 구조 및 항목별 설명 >



항 목	설 명
Prev Hash	이전 블록의 해시 값을 지칭(Parent Hash)
Nonce	작업증명 시 이용되는 64비트 해시로써, 충분한 양의 계산을 위해 이용
Timestamp	유닉스 time() 함수의 출력 값으로 생성 시간을 의미
Uncles Hash (ommer[1]shash)	블록의 엉클 목록의 SHA-3 해시(256비트)
Beneficiary	블록의 채굴 성공 시 모든 수수료(Fees)가 전송되는 160비트 주소
Logs Bloom	블룸필터(Bloom filter)[2]는 거래 목록의 각 거래 영수증에서 각 로그 항목에 포함된 색인 정보(로그 주소 및 주제를 기록)로 구성

Difficulty	블록의 난이도에 해당되는 값으로 이전 블록의 난이도 및 타임스탬프로부터 계산
Extra Data	블록과 관련된 데이터를 포함하는 임의의 바이트 배열
Block Num	상위 블록의 수
Gas Limit	블록 당 가스 비용을 제한하는 값
Gas Used	블록에서 트랜잭션에 사용된 총 가스 값
Mix Hash	충분한 양의 계산을 위해 블록에 수행되는 넌스와 작업 증명을 하는데 이용되는 256비트 해시
State Root	모든 트랜잭션이 실행된 후 완결 짓는데 적용되는 상태 트리의 루트 노드(SHA-3 해시 값)
Transaction Root	블록의 거래 목록 부분에 각 트랜잭션으로 채워진 상태 트리의 루트 노드(SHA-3 해시 값)
Receipt Root	블록의 거래 목록 부분에 각 거래 영수증으로 채워진 상태 트리의 루트 노드(SHA-3 해시 값)

□ 이더리움 특징

o 비트코인과 같이 화폐로서의 교환 기능뿐만 아니라 프로그램 실행을 위한 기능이 추가되었으며, 또한 기능 개선을 위해 일부 수정된 기능이 포함됨

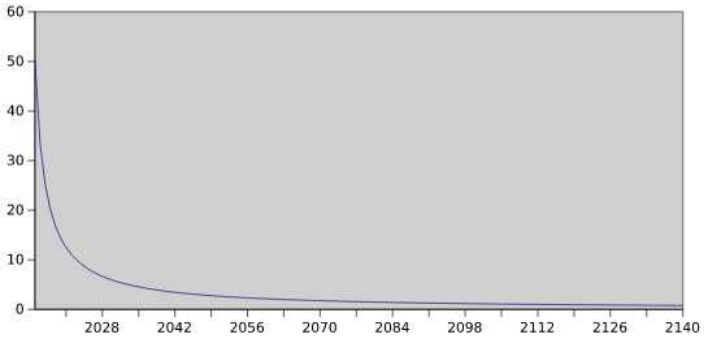
o **(계정)** 비트코인과 동일하게 개인키에 의해 통제되는 외부 소유 계정(Externally Owned Accounts)과 계약 코드에 의해 통제되는 계약 계정(Contract Accounts) 2가지로 구분됨

o **(통화 발행)** 초기에 약 72만 이더(Ether)[3]를 발행하여 투자비용을 위해 사전 판매(62만 이더)하였으며, 총 이더에 대한 신규 이더의 발행률 비중이 '0'이 되도록 매년 줄어들게 함

- 총 통화 발행량은 고정적이지 않으나 인플레이션 등을 고려하여 이더리움 재단(Ethereum Organization)에서 발행량을 유동적으로 조절

1) ommer는 "부모의 형제"를 의미하는 가장 널리 퍼진 성별 중립적인 용어로 Uncle로 주로 쓰임
2) 불름필터 : 데이터 존재의 유무를 알기 위한 확률적 데이터 자료구조
3) 비트코인에 사토시(satoshi)의 개념과 유사하게 소액결제, 수수료 등 작은 단위를 표현하기 위해 1 finney(10^{-3}), 1 szabo(10^{-6}), 1 Gwei(10^{-9}), 1 Mwei(10^{-12}), 1 Kwei(10^{-15}), 1 wei(10^{-18}) 등 이 존재함

< 화폐의 장기 공급 성장률(%) >



o **(채굴)** Ethash라 불리는 수정된 작업증명(Proof-of-Work, PoW) 방식을 이용하며, 약 12초당 한 개의 블록이 생성될 수 있도록 알고리즘이 설계됨

- 채굴에는 DAG 파일(2차원 배열 데이터)이 이용되며, 이것은 GPU[4] 연산을 더 효율적으로 높이고 ASIC[5]을 이용한 채굴을 방지하도록 설계
- 하지만 고속으로 생성되는 블록으로 인해 생성되는 엉클(혹은 Stale)블록이 보안성을 저하*시키므로, 이를 해결하기 위해 수정된 GHOST(Greedy Heaviest Observed Subtree) 프로토콜 도입

* 이중지출(Double-Spending)과 같은 보안 문제 발생 가능

참고 : GHOST 프로토콜

□ 블록들이 네트워크에 전파되는데 일정한 시간이 걸리기 때문에 채굴에 성공한 직후 다른 채굴자가 또 다른 블록을 채굴했다면 결국 낭비되어 네트워크에 기여하지 못함

□ 네트워크 보안 손실 문제를 해결하기 위해 블록의 모(Parent)블록과 조상(Ancestors) 블록, 그 블록의 엉클(Uncle)블록까지 더하고, 수수료를 제외한 기본 보상을 제공함

- 엉클블록은 기본 보상의 87.5%를 받으며, 추가로 엉클블록을 포함하면 12.5%를 받도록 설계

□ 이더리움에서는 7 단계 레벨만 포함하는 단순화된 GHOST 버전으로 구현

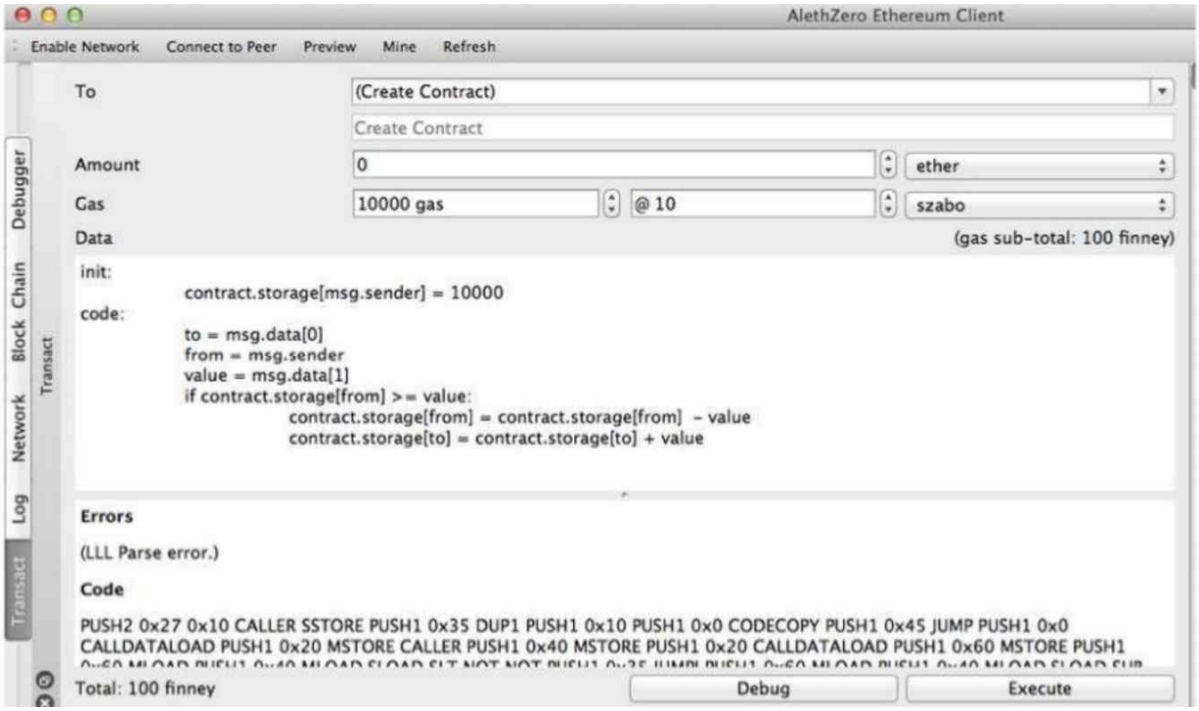
- 하나의 블록은 반드시 하나의 모블록을 지정하며, 0 또는 그 이상의 엉클블록을 지정
- 블록 B에 포함된 엉클블록은 다음과 같은 속성들을 가지고 있어야 한다.
  - B의 k번째 조상의 직접적인 자손이어야 한다. 여기서 '$2 \leq k \leq 7$'.
  - B의 조상이어서는 안 된다.
  - 유효한 블록 헤더여야 하지만, 이전에 확인되었을 필요나 유효한 블록일 필요가 없다.
  - 이전 블록들에 포함된 모든 엉클블록들 그리고 같은 블록에 포함된 모든 다른 엉클블록들과는 달라야 한다(중복포함방지)
- 블록 B에 있는 각 엉클 U에 대해, B의 채굴자는 코인베이스 보상에 더해 추가로 3.125%를 더 받고, U의 채굴자는 기본 코인베이스 보상의 93.75%를 받는다.

4) GPU(Graphic Processing Unit) : 그래픽처리를 위한 고성능의 처리장치
5) ASIC(Application Specific Integrated Circuit) : 반도체 업체가 사용자의 주문에 맞춰 설계·제작해 주는 주문형 반도체

o **(스마트 계약)** 미리 프로그래밍된 규칙에 따라 자동으로 실행되도록 구현된 것으로, EVM(Ethereum Virtual Machine)[6] 코드로 작성됨

- 개발툴로는 AlethZero, MIX 등이 존재하며, 이더리움 관련 프로젝트로 함께 개발 진행 중

< AlethZero를 이용한 구현의 예 >



o **(분산어플리케이션[7])** 이더리움의 EVM에서 동작하는 분산환경기반 응용 프로그램으로써, 대표적인 예로 비트토렌트(BitTorrent)*, 비트코인 등이 존재함

* P2P 시스템에 의해 네트워크가 유지되며, 개인간 파일 전송 프로토콜 제공

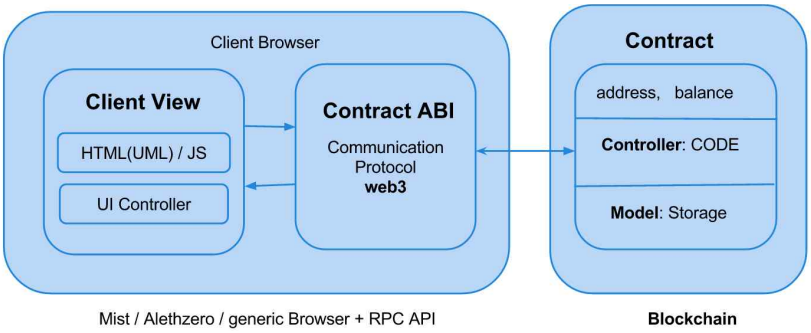
- 이더리움 네트워크의 자원을 활하여 신뢰성, 안전성을 제공하고, 서버 운영 및 보안을 위한 추가 비용이 발생하지 않으나 프로그래밍 코드가 복잡할수록 가스 사용량이 증가하여 수수료 부담을 증가시킴
- 주식, 채권, 보험, 복권, 도박, 토큰, 쿠폰, 투표, 기록, 에스크로(Escrow), 예측 시장(Prediction Market) 등에 활용 가능
- 분산어플리케이션은 이용자의 브라우저에서 이더리움에 저장된 계약 코드를 실행하기 위한 기능*을 제공

6) 자바 가상머신(JVM, Java Virtual Machine)과 유사한 개념으로 C++, 자바스크립트, 파이선, GO 등 4가지 지원 언어로 작성된 코드를 컴파일에 의해 바이트 코드로 변환되면 이를 해석하여 실행하는 소프트웨어를 일컬음

7) Dapps, Distributed Applications

* Contract ABI(Application Binary Interface)는 이기종 컴퓨터 상호 간에 응용 프로그램의 이식성을 실현하기 위해 2진 호환(기계어로 작성된 프로그램이 다른 기기에서도 작동하는 성질)을 보증하는 인터페이스 규약

< 분산어플리케이션의 구조 >



□ 시사점

o **(환경 변화)** 이더리움은 자유로운 웹 생태계를 제공함으로써 새로운 인프라 조성에 기여 할 것으로 예상되며, 초연결사회(Hyper-Connected Society)에서 보다 효과적으로 이용될 수 있을 것으로 보임

- 국경의 경계가 무의미해짐에 따라 국내 기업이 해외로의 진출이 용이하나, 기술을 선도하는 글로벌 기업에 의해 독점될 수 있음
- 또한, 중개인 역할을 대체함으로써 직업의 변화에도 영향을 줄 것으로 예상되므로 사회적 부작용에 대해서도 검토 필요

o **(지속적 실험)** 기존 비트코인에서의 비효율적인 부분(작업 증명 등)을 개선시키기 위해 다양한 실험이 진행되고 있으며, 기능 개선을 위한 업데이트가 예고됨에 따라 지속적인 관심이 필요함

- 4단계로 나누어 서비스를 출시할 예정이며, 최근 Homestead를 발표

※ 1) Frontier, 2015년 4월 출시
2) Homestead, 2016년 2월 말 출시
3) Metropolis, 미정
4) Serenity, 미정

o **(추가 고려사항)** 익명성을 악용한 불법 거래 수단(마약/총기거래, 자금세탁 등)으로 이용될 수 있으며, 화폐 발행에 제한이 없어 과잉 공급으로 인한 문제점 등 사전 대비가 필요함

[참고자료]

[1] Ethereum Wiki, https://github.com/ethereum/wiki/wiki

[2] Ethereum White Paper, https://github.com/ethereum/wiki/wiki/White-Paper

[3] Ethreum Yellow Paper, http://gavwood.com/Paper.pdf

[4] 비트코인 Bitcoin, 이더리움 Etheruem, 경제/일상이야기,
http://www.seunghwanhan.com/2015/07/2015-07-17-ethereum-status-now.html

[5] 비트코인과 블록체인 2.0,
http://atomrigs.blogspot.com/2014/11/ethereum.html?view=sidebar