ETRI

2017
Electronics and
Telecommunications
Trends

비트코인 후 블록체인

Blockchain Beyond Bitcoin

허세영 (S.Y. Hong) 정보보호연구본부 연구원
조상래 (S.R. Cho) 정보보호연구본부 책임연구원
김수형 (S.H. Kim) 정보보호연구본부 책임연구원/실장

2008년, Satoshi Nakamoto라는 가명을 쓰는 신원미상의 사람(들) 혹은 단체가 비트코인을 소개하였다[1]. 그 이후, 현재 비트코인의 총액은 10억달러 이상에 달한다[2]. 비트코인을 지탱하는 기술인 블록체인에 의해 이론적으로 비트코인은 거의 조작이 불가능하며 거의 영구적으로 기록에 남게 된다. 이 획기적인 기술로 인해 비트코인 이 후 약 700개가 넘는 암호화폐가 생겨났으며[2], 암호화폐뿐만이 아닌 금융권/비금융권, 정부기관 등 많은 도메인에서 블록체인 기술이 도입되어 가고 있다[3]-[5]. 비트코인 이후 가장 대표적인 블록체인 기반의 프로젝트로는 튜링 완전한 컴퓨팅 기능이 있는 이더리움이 있다[6]. 이 튜링 완전한 컴퓨팅 기능으로 인해 이더리움 상에서 투명하게 데이터 저장 및 프로그램 실행을 할 수 있다[6]. 이로 인해 이더리움을 이용한 많은 혁신적인 이용 사례가 생겨났다[7]. 본고에서는 비트코인 후 블록체인의 기술을 이더리움 중심으로 소개하며 해결해야 할 기술적 이슈들과 동향에 관해 분석한다. 그리고 현재 이더리움 외에 개발되어오는 블록체인에 대해 간략히 살펴본다.

OPEN KOGL 공공저작물 자유이용허락
본저작물은 공공누리 제4유형
출처표시+상업적이용금지+변경금지 조건에 따라 이용할 수 있습니다.

초연결 지능 인프라 특집

© 2017 한국전자통신연구원

Ⅰ. 서론

블록체인, 비트코인, 가상화폐 등 많은 사람들이 한두 번씩은 들어봤을 법한 단어들이다. 이론적으로 '거의' 조작이 불가능하며 저장된 기록이 '거의' 영구적으로 남을 수 있다는 특성[8]이 있기에 보안학계를 포함한 많은 학계 그리고 업계에서 화두가 되고 있다.

이로 인해 블록체인은 가상화폐는 물론 다양한 분야의 산업에서 접목되어 가고 있다[3]-[5]. 정부기관, 스타트업, 그리고 대기업 등 다양한 주체에서 이용되고 있으며 보안 분야에서도 활발히 연구 및 활용이 되고 있다[3]-[5]. 특히 IBM이나 Microsoft에선 클라우드 서비스 혹은 IoT(Internet of Things)와 블록체인을 접목시키려는 시도가 되고 있다[4][9].

비트코인 이후 많은 블록체인 기반의 전자화폐가 생겨났으며 대표적으로 이더리움이 있다. 이더리움은 비트코인의 단점들을 보완하고 튜링 완전한 컴퓨팅 기능을 넣음으로써 블록체인상에서 프로그램을 실행시킬 수 있는 기능을 더하였다[6].

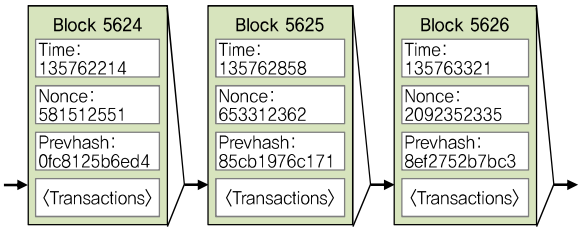
그래서 본고에서는 비트코인 후 블록체인, 특히 이더리움에 관해서 살펴보자고 한다. 이더리움과 비트코인의 차이점, 이더리움의 작동방식, 이더리움의 실제 활용사례 등을 설명하려고 한다. 비트코인 이후 블록체인에 관해 설명하고자 하므로 비트코인에 관한 동향 및 설명에 관해 알고자 하면 블록체인 기술개념 및 적용현황[10]을 참조하면 된다.

Ⅱ. 비트코인 기반의 블록체인

1. 비트코인

이더리움을 살펴보기에 앞서 간략하게 비트코인 기반 블록체인의 기본적인 특성을 설명하려고 한다. 블록체인은 아래 (그림 1)과 같이 블록으로 연결되어 있다.

블록은 주기적으로 생성되며 (그림 1)과 같이 블록체인에 추가된다. 예를 들어 블록 5625 다음 생성된 블록은 블록 5626이 되며 블록 5625와 연결된다. 비트코인의 경우 대략 10분에 한 블록이 생성되며 이더리움의 경우 대략 12.4초에 한 블록이 생성된다[11].



(그림 1) 블록체인 Diagram

여기서 주목해야 할 부분은 블록이 포함하는 요소들이다. 시간과 난수, 전 블록의 해쉬, 그리고 거래내역들 등을 포함한다.

가. 시간

블록이 생성된 시간을 의미하며, 시간을 기록함으로써 공격자가 해킹 못하게 하는 용도로도 쓰인다. 일반적으로 Unix time으로 표현된다[12].

나. 난수

비트코인의 경우 4바이트값으로서 SHA256을 이용하여 해쉬값을 조정하는 데 쓰인다. 예를 들어 난수의 1비트가 바뀌어도 SHA256의 결과값이 달라지기 때문에 채굴자들이 채굴해서 얻으려는 해쉬값을 얻으려면 채굴자는 많은 조합의 난수를 대입해 보아야 한다[12].

다. 전 블록의 해쉬

블록을 가리키며, 전 블록을 가리켜야 블록 간의 연결이 되기 때문에 필요하다[12].

라. 거래내역들

한 블록이 생성될 때 거래내역들도 같이 포함된다[12]. 예를 들어 블록 5626이 생성된 후 x개의 거래량이

있다면 Transaction Pool에 저장이 되며, 블록 5627가 생성되고 블록체인에 포함될 때, 블록 5627의 거래내역들에 포함된다. 더 자세히 설명하자면 채굴의 답을 얻을 때 비트코인의 경우 SHA256으로 원하는 값을 얻는데, SHA256에 포함되는 입력 값 중 하나가 거래내역에 관한 데이터이다. 정확하게는 SHA256에 들어가는 거래내역과 연관된 값은 Merkle Tree를 이용한 값이다. 그러므로, 이론적으로 채굴자가 채굴의 답을 찾을 때 일부 거래내역들을 포함하지 않고 SHA256을 계산했을 때엔 블록이 생성될 때 그 일부 거래내역들이 포함이 되지 않는다. 그리고 일반적으로 블록의 크기가 정해져 있기 때문에 짧은 시간 내에 너무 많은 거래가 요청된다면 한 블록에 모든 거래내역들이 포함 안될 수도 있다. 포함되지 않은 거래내역들은 추 후에 생성된 블록에 포함된다[13]. 하지만 단순화를 위해 그런 부분들은 생략한다. 또한, 블록체인의 특성상 '51% Attack' 이나 'Selfish Miner Attack' 등의 공격[13]으로 인해 이론적으로 완벽히 영구적이거나 조작 불가능하지는 않기 때문이 '거의' 영구적이고 '거의' 조작 불가능하다고 서술한다.

2. 비트코인의 단점

비트코인은 블록체인의 가장 좋은 예이기도 하지만 몇 가지 단점들[6]이 존재한다.

가. 튜링 미완전성(Turing Incompleteness)

단순하게 말해서 while문과 for문 등을 이용할 수가 없다. 실제로 비트코인에서도 스크립트 기능이 존재한다. 하지만 일반적인 Shell이나 Perl 스크립트와 같다고 할 수 없는 이유는 for문이나 while문 등의 되풀이되는 기능을 이용할 수 없다는 데에 있다. 그래서 비트코인 블록체인에서 허가된 단순한 명령 정도만 실행시킬 수 있다. 이런 연속되는 작업을 하기 위한 명령을 이용함으로써 장점이 많지만 비트코인에서 이 기능을 넣지 않은 이유가 존재한다.

비트코인을 만약 철저한 디자인이 없이 이 기능을 구현하였다면 아주 간단하게 DOS(denial of service) 공격을 할 수 있었을 것이다. 예를 들자면 공격자가 스크립트에 while(1){}을 삽입하고 비트코인이 이 스크립트를 실행한다면 단순하고 의미 없는 무한 작업이 블록체인에서 실행될 것이다. 그리고 단시간에 비트코인 네트워크는 마비가 될 것이다.

나. 가치의 불투명성(Value Blindness)

비트코인을 화폐로 본다면 그 가치를 확실히 알 수 있어야 한다. 예를 들어 x 달러를 y 비트코인으로 환전했다고 가정하자. 30일 뒤에 y 비트코인을 달러로 환전하려고 할 때 y 비트코인과 상응하는 달러의 가치를 알아야 한다. 하지만 y 비트코인의 가치를 쉽게 알 수 있는 방법이 존재하지 않는다. 물론 사람이 개입하여 일일이 확인을 해볼 수는 있지만 자동으로 가치를 알 방법은 존재하지 않는다.

다. 상태의 단순성(Lack of State)

비트코인의 경우 사용된 비트코인과 사용하지 않은 비트코인, 즉 두 가지 상태로만 나뉜다. 하지만 실제 상황에선 수많은 상태들이 존재할 수 있다. 할부의 개념도 존재할 수 있고, 조건이 주어지고 그 조건에 충족이 되었을 때만 비트코인을 이용해야 하는 상황도 존재할 수 있다. 하지만 비트코인으로는 이 많은 경우를 전부 구현하기 힘들거나 불가능하다.

라. 검증 시간

앞서 설명했듯이 비트코인의 경우 한 블록이 생성되는데 대략 10분의 시간이 소요된다. 한 블록이 생성되었을 시, 한 검증이 되었다고 말하는데, 검증의 수가 늘어날수록 거래의 신용도도 높아진다. 예를 들어 3개의

검증보다 10개의 검증을 더 신뢰할 수 있다. 큰 거래를 할 시 대략 한 시간 정도를 기다릴 수 있지만, 커피 한잔을 사는 데 한 시간을 기다리는 건 상식에 어긋난다.

Ⅲ. 이더리움 기술의 설명

위에 언급한 비트코인의 문제점들을 이더리움 기반의 블록체인은 쉽게 해결을 한다. 이더리움의 특징에 관한 설명에 앞서 이더리움의 역사에 대해 살펴보겠다.

1. 이더리움의 역사

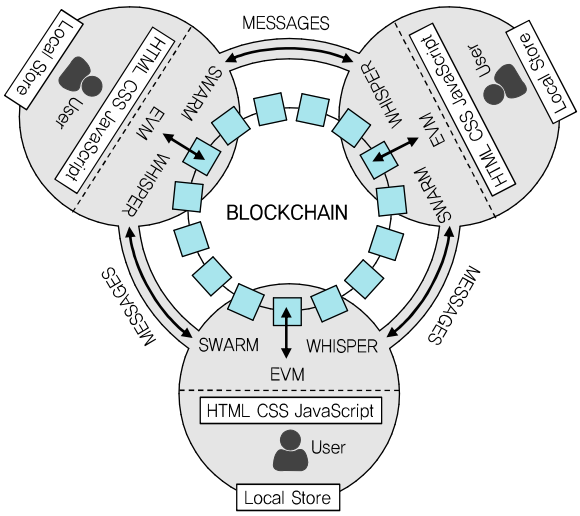
이더리움은 2013년 Thiel Fellow인 Vitalik Buterin에 의해서 제안되었다[14]. 2015년 중순, 첫 공개가 된 후, 많은 찬사를 받으며 단기간에 두 번째로 규모가 큰 가상화폐로 자리매김을 했다[10].

비트코인이나 타 가상화폐와는 다르게 스마트 컨트랙(Smart Contract)이라는 개념을 접목했다는 부분이 이더리움의 성공에 큰 영향을 주었다. 1994년 암호학자 Nick Szabo에 의해 처음으로 제안된 스마트 컨트랙을 블록체인에 접목시킴으로써 이더리움은 기본적인 거래 장부 기록 외에 튜링 완전한 컴퓨팅 기능과 그 기능을 이용하여 프로그램을 실행할 수 있는 환경을 제공하였다[11][14].

하지만, 이더리움은 소개된지 3년도 되지 않아 엄청난 성공을 했지만 DAO Attack, OPCODE Computational DDOS Attack등의 공격을 당하기도 하였고 타 블록체인에 비해 복잡한 구조로 인해 아직까지도 많은 공격을 받고 있다.

2. 이더리움

이더리움의 경우 비트코인의 단점을 보완하면서 단순히 A에서 B로의 거래를 기록하는 것이 아닌 컴퓨팅 기능을 추가함으로서 큰 의미를 가진다. 이더리움 개발자들은 이더리움을 'Blockchain with a built-in Turing-complete programming language' 라고 정의한다[6]. 즉, 블록체인에서 프로그램을 구현할 수도 있고 실행시킬 수도 있다는 특성이 있다. 그래서 이더리움은 (그림 2)와 같이 비트코인보다는 복잡한 구조를 가지고 있다.



(그림 2) 이더리움 구조도

〈출처〉: Steemit.com

가. 이더리움 Accounts

이더리움도 비트코인의 지갑과 동일한 개념인 account가 존재한다. 하지만 비트코인과는 다르게 두 가지 종류의 account가 존재한다.

- Externally Owned Account(EOA)
  비트코인의 지갑과 같은 개념이다. 주소가 존재하며, 이더리움의 통화단위인 이더(Ether) 등을 포함한다. ECDSA기반으로 공개/비밀키가 만들어진다.
- Contract Account
  이더리움의 계정 중 하나이다. 일반적으로 EOA의 경우 이용자의 지갑과 동일하게 이용되지만 Contract Account는 블록체인 내부에 존재하며 몇 가지 특성을 가지고 있다. 첫번째로 코드를 포함할 수도 있으며 블록체인 외부 혹

은 내부에서 코드를 실행시킬 수 있다. 두번째로 데이터를 저장할 수 있다. 그리고 다른 EOA와 마찬가지로 이더를 소유할 수 있다.

나. 이더리움의 컴퓨팅 기능

이더리움 블록체인의 경우 이더리움 버추얼 머신(Ethereum Virtual Machine)이 존재한다[6]. 실제로 이더리움에는 Solidity와 Serpent등[15][16]과 같은 프로그래밍 언어가 있으며 그 언어를 이용하여 프로그램을 구현할 수 있다. 그 프로그램을 구현한 뒤 컴파일을 하면 바이트코드로 변환된다. 그리고 그 바이트코드에 대응되는 연산 부호가 있다. 예를 들어 STOP의 경우 0×00, ADD의 경우 0×01으로 해석된다.

또 하나 흥미로운 점은 튜링 완전한 컴퓨팅 기능에 있다. 위에서 언급한 것과 같이 while(1){}같은 코드를 실행하게 된다면 DOS 공격을 단순하게 할 수 있다. 블록체인의 특성상 참여한 채굴자가 코드를 검증 및 실행하게 되므로 위와 같은 코드가 참여한 모든 기계에서 실행되었을 시엔 블록체인 자체가 마비될 수가 있다. 하지만 이 문제를 해결하기 위해 가스(gas)의 개념[6]이 존재한다. 연산부호마다 가스 가격이 있으며 처음 코드를 실행시킬 때 가스 양을 정할 수 있다. 예를 들어 10000 가스를 이용하여 프로그램을 실행시킬 시, 가스가 부족하면 코드를 끝까지 실행시킬 수 없게 된다. ADD의 경우 3 가스, MUL의 경우 5 가스 등으로 책정되어있다. 그래서 while(1){}과 같은 코드를 실행하려고 한다면 그만큼 천문학적인 혹은 무한한 양의 이더를 소모하여야 한다.

다. 튜링 완전한 컴퓨팅 기능의 장점

앞서 언급했듯이 블록체인은 '거의' 위변조 불가능하며 기록이 '거의' 영구적으로 남을 수 있다는 특징이 있다. 일반적인 분산원장이 아닌 이더리움과 같이 프로그램을 블록체인에서 실행함으로써 얻는 많은 장점이 있다.

첫 번째로 이더리움상에 코드를 신용할 수 있다. 일반 서버에 작동하는 코드들은 실제 서버 내에서 어떤 일이 일어나는지 완벽하게 알 수 없다. 예를 들어 서버에서 투표나 도박을 하는 서비스가 존재한다면 그 서비스를 완벽히 신용할 수는 없다. 하지만 모든 코드와 실행되는 단계가 블록체인 참여자들에게 공개가 된다면 이용자들은 그 서비스를 신용할 수 있게 된다.

두 번째로 한번 저장된 코드 및 데이터는 '거의' 영구적으로 남는다. 그러한 이유로 인해 변조가 되면 안 되는 데이터를 블록체인에 저장하는 등의 용도로도 활용할 수 있다.

그리고 위 예시와 같이 이더리움 블록체인에서 작동하는 프로그램을 decentralized app혹은 dAPP이라고 부른다.

라. 이더리움 샘플 코드 예시

(그림 3)은 이더리움상에서 작동할 수 있는 간단한 코드이다[6]. 위 코드를 설명하기 앞서 변수 및 환경을 설명하겠다.

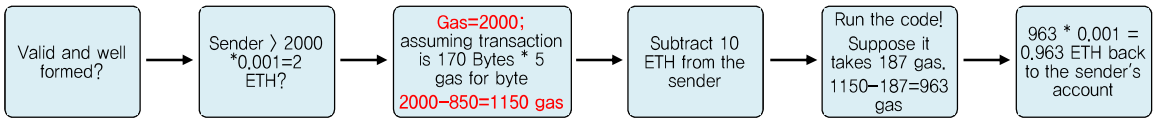
- self.storage: 데이타가 저장되는 스토리지로서 처음엔 데이타를 포함하고 있지 않음.
- calldataload: 0-31 byte까지는 숫자 2, 그리고 byte 32-63까지는 'CHARLIE'를 포함

이 코드를 실행하기 위해 EOA는 10이더를 보유하고 있으며, 2000 가스를 이용하며, 0.0001 이더 가스 가격(1 가스당 0.0001 이더 소모) 비율로 가스가 사용된다.

첫 번째 줄의 self.storage[2]에는 데이터가 존재하지 않기 때문에 두 번째 줄의 코드가 실행된다. 그리고 self.storage[2]에 'CHARLIE'가 저장된다. 간단한 코드

```
if !self.storage[calldataload(0)]:
    self.storage[calldataload(0)] = calldataload(32)
```

(그림 3) Serpent Sample Code



(그림 4) 실제 소스코드 실행시 단계

지만 이더리움 내부에서는 여러 가지 일들이 실행된다. (그림 3)의 코드를 실행했을 시, (그림 4)와 같은 단계를 따른다. 먼저 코드를 실행한 이용자를 확인한다. 예를 들어, EOA가 올바른 포맷을 사용하였는지, 충분한 이더를 소유하고 있는지, 등을 블록체인의 참여자들이 확인을 한다. 그 후, 코드를 이더리움 네트워크에 저장을 해야 한다. 코드의 용량을 기준으로 책정이 되기 때문에 코드의 용량이 170 바이트이며, 한 바이트 당 5 가스가 사용된다고 가정한다면, 170 와 5를 곱해야 하기 때문에 850 가스가 사용된다. 2000 가스에서 850 가스가 사용되므로 1150 가스가 남는다. 위 코드를 실행하는 데 187 가스가 든다고 가정할 땐 1150가스에서 187가스를 뺀 963 가스가 남는다. 남은 963 가스는 0.0001 이더가 1가스라고 가정했을 때 0.963 이더가 되기 때문에, 0.963 이더는 다시 코드를 실행한 EOA로 반환된다. 위에서 설명하였듯이 소스코드는 컴파일을 시키면 바이너리가 되며 그 바이너리와 상응하는 연산부호가 존재한다. 기계 언어의 연산부호와 같다고 생각하면 된다. 그리고 기계언어의 연산부호에도 클락 사이클 등에 따른 비용이 있는 것처럼 이더리움 연산부호도 각 연산부호마다 가스의 가격이 정해져 있다.

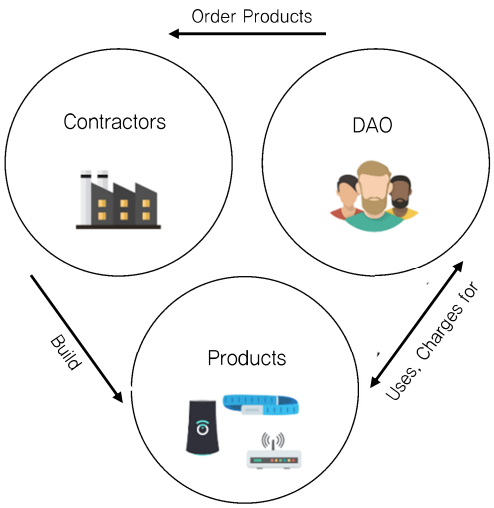
위에 언급한 것과 같이 가스가 존재하기 때문에 DOS 공격과 같은 문제를 쉽게 해결하였다. 첨언하자면 이론적으로는 해결하였지만 실제상황에서는 완벽히 해결하진 못하였다.

3. 이더리움의 취약점을 이용한 공격

이더리움은 블록체인 기반의 컴퓨팅 머신을 구현하여서 혁신적인 변화를 불러왔지만 그만큼 많은 공격을 당하였다. 피해 규모 면에서 가장 이슈가 된 2건의 공격이 있다.

가. DAO Attack

2016년 4월, 이더리움 재단에서는 이더리움의 장점을 살려 크라우드 펀딩을 시작하였다. The Decentralized Autonomous Organization(The DAO)라고 칭하는 스마트 컨트랙을 구현하여서 투명하게 펀딩을 받으며, 주식회사에 투자하는 방식과 비슷하게 투자하여 지분을 받는 방식으로 구현하였다. DAO라는 통화단위를 이용하여 투자자들은 투자한 액수만큼 DAO를 소유할 수 있게 하였다. 비지니스 모델은 아주 간단하면서 혁신적이었다. (그림 5)에서 보듯이 DAO를 소유한 투자자들이 투자를 받고자 하는 사업체의 제안서를 검토하고 DAO를 투자하는 방식이다. 투자를 받은 사업체는 투자액을 이용하여 사업을 하고 이윤을 다시 투자를 한 투자자에게 줄 수 있다. 이더리움의 투명성을 이용하여 투자자들은 자신이 투자한 돈의 유동을 쉽게 알 수 있게



(그림 5) The DAO

〈출처〉: forum.daohub.org

된다는 장점이 있다. 실제로 이 크라우드 펀딩은 엄청난 성공을 거두어서 6월 초 한화 기준으로 약 2천억원이 넘는 돈을 모았다. 크라우드펀딩 역대 최고액을 갱신하며[17] 시장의 큰 기대를 받았지만, 6월 18일, 해커들에 의해서 The DAO에 보관되어 있는 360만개 이상의 이더들이 해커의 계정으로 옮겨졌다[18]. 이 사건으로 인해서 당시 1 이더의 가치는 20달러에서 이틀만에 13달러로 하락했다. 실제로 공격은 이더리움의 내재된 취약점이라기 보단, 이더리움 기반의 프로그램 즉 스마트 컨트랙 코드의 결함이었다[18]. 단순히 설명하자면 이더를 A에서 B로 옮겨주는 단계에서 옮겨진 값을 바로 수정하지 않는 취약점으로 인해 재귀함수 공격에 취약점이 있었다. 이 공격으로 인해 이더리움 재단에서 나설 수 밖에 없게 되었고 재단이 이더리움 네트워크에 직접적으로 개입함으로써 탈 중앙화를 이더리움의 장점이라고 하던 재단의 행보에 문제가 생길 수 밖에 없었다. 그리고 hard fork, 이더리움 블록체인을 업데이트, 함으로서 찬반이 나뉘게 된다[19]. 결과적으로 이더리움은 hard fork로 인해 ETH진영과 ETC 진영으로 나뉘게 된다.

나. OPCODE DDOS

이더리움 코드는 Solidity나 Serpent라는 프로그래밍 언어로 구현되며 그 코드가 바이트코드로 변환된다. 일반적인 X86기반과 비슷하게 연산부호가 존재하며 바이트코드는 연산부호의 인스트럭션에 따라서 실행된다. 일반적인 컴퓨터 구조와 같이 스택, 메모리, 프로그램 카운터 등이 모두 존재한다. 위에 언급한 것과 같이 ADD, MUL, SUB, CREATE 등의 연산부호들이 존재하며 연산부호마다 가스 비용이 있다.

예를 들어 계산이 많은 연산부호 같은 경우 대체적으로 가스 비용이 높다고 예상할 수 있다. 하지만 많은 연산부호가 존재하기 때문에 모든 연산부호에 적절한 가스 비용을 부여하긴 어렵다.

그 취약점을 이용하여 공격자들은 EXTCODESIZE라는 연산부호로 이더리움 네트워크를 공격[20]하였다. EXTCODESIZE의 경우 비교적 저렴한 가스 비용이 들지만 디스크에서 데이터를 읽어와야하기 때문에 계산 비용이 큰 연산부호이다. 그렇기 때문에 공격자는 저렴한 가스 가격의 EXTCODESIZE를 실행시킴으로서 이더리움 네트워크를 마비시켰다. 실제 실험결과 30~60초 내로 실행이 되어야 할 프로그램이 5시간 이상 지체되기도 하였으며 결재가 취소된 적도 있었다. 이 사건으로 인해서 이더리움 재단에서는 또 다시 hard fork를 감행하였다[21].

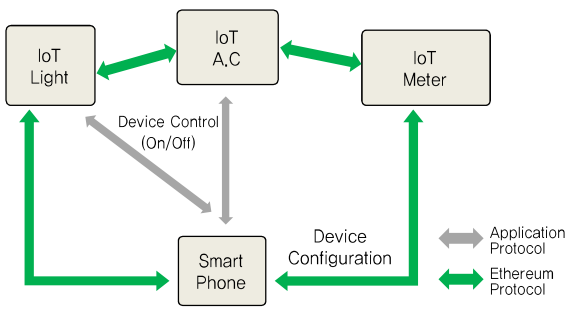
Ⅳ. 이더리움의 이용사례 및 동향

이더리움은 스마트 컨트랙, 즉 프로그램을 저장하고 실행시킬 수 있는 특징이 있으므로 많은 응용 서비스에서 이용될 수 있다. 그래서 ETRI에서는 이더리움을 이용하여 IoT 기기 인증 및 제어하는 시제품을 구현해 보았다.

1. 이더리움을 이용한 IoT 기기인증 및 제어

가. 구성 및 동작 방법

라즈베리 파이를 이용하여 전기소모량을 모니터링하는 스마트 미터, 스마트 에어컨, 그리고 스마트 전구를 (그림 6)과 같이 시뮬레이션하였다. 스마트 미터는 (그



(그림 6) 블록체인을 이용한 IoT 기기인증

```
contract SmartMeter_Demo{
        int meter;
        bytes pubKey;
        bytes data_sign;

        function meter_data(int _meter, bytes _publicKey,
bytes _signature){
              meter = _meter;
              pubKey = _publicKey;
              data_sign = _signature;

              PrintData(meter, pubKey, data_sign);
        }

       ...More Code...
  }
```

(그림 7) Smart Contract 예

림 7)의 코드를 이용하여 실시간으로 미터 값(meter), 공개키(pubKey), 데이터 사인(data_sign)값을 블록체인에 기록하였다. 그리고 스마트폰을 사용하여 이용자가 에어컨과 전구의 정책을 설정한다. 예를 들어 전기 사용량이 300KW가 넘을 시 에어컨을 절전모드로 바꾸는 등의 정책을 설정할 수 있다. 이용자가 에어컨의 절전점을 300KW로 블록체인에 설정해놓으면 에어컨 기기는 실시간으로 미터기의 값과 에어컨 설정값을 비교한다. 여기서 에어컨 기기는 미터기의 값을 공개키와 데이터 사인값을 이용하여 인증할 수 있다. 만약 공격자가 임의의 데이터를 블록체인에 기록한다면 기기는 데이터가 잘못되었다는 것을 공개키와 데이터 사인값을 확인하여 알 수 있다.

나. 시사점

블록체인을 이용하여 IoT 기기를 제어해보면 수많은 IoT 기기들을 블록체인으로 연결하여 관리하면 효율적일 거라고 예상할 수 있었다. 예를 들어 IoT 기기들이 기하급수적으로 늘어날 때 서버-클라이언트의 구조보다 훨씬 더 확장성에서 유용할 것으로 예상된다. 그리고 많은 기기들이 서로 동기화하는 부분에서도 큰 장점이 있다.

하지만, 기술적으로 해결하여야 할 부분들도 분명히 존재한다. 첫 번째로 거래시간에 대한 문제이다. 현재 이더리움의 경우 한 검증 시간당 12초 정도가 걸린다. 비트코인보다 짧은 시간이지만 시간에 민감한 도메인에서는 몇 십 초를 기다리는 데엔 무리가 있다.

두 번째로 거래시간의 불확실성이다. 이더리움의 검증시간을 12초 정도로 예상을 하고 있지만 완벽하게 확신을 할 수는 없다. 최근 있었던 Computational DOS Attack과 같은 공격이 있다면 얼마든지 네트워크는 마비될 수 있으며 그렇게 된다면 검증 시간을 확실히 알 수 없게 된다.

또한, 비트코인이나 이더리움 등과 같은 퍼블릭 블록체인을 이용할 경우 비용이 발생할 수밖에 없다. 블록체인에 용량이 큰 데이터를 주기적으로 저장한다면, 많은 비용이 발생할 수도 있다.

위 문제점들을 해결하기 위해서는 컨소시엄 블록체인이나 다른 형태의 블록체인을 고려해야 할 필요가 있다.

V. 그 외의 블록체인 기반의 기술

지금까지 이더리움에 대해 살펴보았다. 이더리움의 획기적인 특성들로 인해 많은 주목을 받고 있지만, 이더리움 외에도 많은 블록체인들이 연구 개발되고 있다. 대표적으로 Hyperledger 프로젝트, R3CEV의 Corda, ZCash에 관해 간략하게 살펴보겠다.

1. Hyperledger

2015년 12월 리눅스 재단에서 시작되었다. 현재 존재하는 대다수의 블록체인은 전자화폐와 접목된다고 볼 수 있다. 이러한 블록체인은 퍼블릭 블록체인으로써 다양한 용도들이 존재하지만 분명히 한계점이 존재한다. 예를 들자면 특정인들만 이용할 수 있는 블록체인과 아무나 이용할 수 있는 퍼블릭 블록체인은 확실히 용도가 다르다고 할 수 있다. 그래서 현존하는 많은 퍼블릭 블록체인의의 한계를 보안하기 위해 만들어졌다. Hyper-

ledger내에 여러가지 프로젝트들이 존재하며, 실제 시간에 민감한 서비스 등의 문제를 해결하기 위해 실시간성을 띄면서 많은 거래를 처리할 수 있는 안정적인 시스템 제작, 컨소시엄 형태의 블록체인 제작 등의 관한 프로젝트 등이 진행 중이다[22]. IBM, Intel, VMWare등과 같은 기술적 배경의 회사들부터 JP Morgan, BNY Mellon와 같은 금융권/은행권 회사들도 참여 중이다.

2. R3CEV의 Corda

R3CEV는 현재 가장 큰 규모의 금융권을 위한 블록체인 컨소시엄이다. 도이체방크, 노무라, 바클레이스, 크레딧스위스 등 50여개 이상의 회사들이 참여하고 있다. 금융권에서 필요한 복잡한 거래들과 기능들을 블록체인을 이용해 단순화 하기 위해 만들어졌으며 최근 위 목적을 달성하려고 Corda라는 프로젝트를 발표하고 오픈소스하였다[23].

3. ZCash

2016년 10월에 오픈된 전자화폐로서 기존 비트코인이나 이더리움 등의 블록체인과는 다른 특징을 지니고 있다. 기존의 퍼블릭 블록체인은 익명성을 보장한다고 주장하지만 계정을 만든 이가 누구인지 알 수 없을 뿐 블록체인 내의 거래내역은 전부 확인할 수 있기 때문에 수신자와 송신자, 그리고 금액은 알 수 있다. 그래서 완벽한 익명성은 보장되지 않는다. 하지만 ZCash의 경우 Zero-Knowledge Proof를 이용하여 수신자와 송신자 그리고 금액을 전부 숨길 수 있다[24].

Ⅵ. 결론

본고에서는 비트코인의 성공 이후, 블록체인의 동향 및 사례에 관해 살펴보았다. 특히 비트코인 이후 가장 화두가 되고 있는 이더리움에 관해 면밀히 살펴보았다. 이더리움은 많은 비트코인의 단점을 보완하며 튜링 완전성을 보장하는 컴퓨팅 기능을 이용하여 일반적인 거래장부 외에도 데이타를 저장할 수 있으며 프로그램도 실행시킬 수 있다.

그로 인해 사물인터넷에서 기기제어나 투표 서비스등과 같이 광범위한 응용서비스에 활용될 수 있다. 이더리움 외에도 많은 블록체인을 이용한 연구개발이 활발히 진행되어 가고 있다. 금융권/비금융권, 정부기관, 기업, 학계 등 많은 분야에서 이용사례가 늘고 있으며, 앞으로 더 많은 이용사례가 생겨날 것으로 예상된다.

화폐나 금융거래에 사용되는 블록체인 기술보다 현재는 실시간으로 동기화가 이루어지는 분산 데이터베이스 특성을 갖는 분산장부 기술쪽으로 발전하고 있다. 분산장부의 '거의' 영구적이며 '거의' 변조가 불가능한 특성으로 인해 비용절감 및 현재 서버 클라이언트로 해결하기 힘든 문제들을 해결할 수 있을 것으로 예상되며 국내에서도 블록체인에 관한 더 많은 연구와 개발이 시급한 실정이다.

참고문헌

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[2] Crypto-Currency Market Capitalizations, https://coinmarketcap.com/ currencies/

[3] McKinsey & Company, "Blockchain in insurance – opportunity or threat?" 2016.

[4] B. Greenstein, "IBM Blockchain Brings Trust and Trans-Parency to the Internet of Things," 2016.

[5] L. Shen, "Blockchain Will Be Used By 15% of Big Banks by 2017," Fortune, 2016.

[6] Ethereum, "Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform," 2016.

[7] Around the Block, "A Current List of Use Cases for Ethereum," 2016, https://medium.com/@AroundTheBlock_/ a-current-list-of-use-cases-for-ethereum-b8caa5807553#.8rt4p9xmy

[8] A. Narayanan et al, "Bitcoin and Cryptocurrency

Technologies," Princeton University Press, 2016.

[9] Microsoft Azure, "Blockchain as a Service," 2016, https://azure.microsoft.com/en-us/solutions/blockchain/

[10] 김영삼 외, "블록체인 기술 개념 및 적용 현황," IITP 주간기술동향, 2016.

[11] V. Buterin, "Toward a 12-second Block Time," 2014, https://blog.ethereum.org/2014/07/11/toward-a-12-second-block-time/

[12] Bitcoinwiki, "Block Hasing Algorithm", https://en.bitcoin.it/wiki/Block_hashing_algorithm

[13] Bitcoin Developer Guide, "Bitcoin Developer Guide," https://bitcoin.org/en/developer-guide

[14] The Cointelegraph, "A Brief History of Ethereum From Vitalik Buterin's Idea to Release," 2015, https://cointelegraph.com/ethereum-for-beginners/a-brief-history-of-ethereum-from-vitalik-buterins-idea-to-release

[15] Solidity, "Solidity", https://solidity.readthedocs.io/en/latest/

[16] Serpent, "Serpent," https://github.com/ethereum/wiki/wiki/Serpent

[17] G. Prisco, "The DAO Raises More than $117 Million in World's Largest Crowdfunding to Date," 2016, https://bitcoinmagazine.com/articles/the-dao-raises-more-than-million-in-world-s-largest-crowdfunding-to-date-1463422191/

[18] D. Siegel, "Understanding the DAO Attack," 2016, http://www.coindesk.com/understanding-dao-hack-journalists/

[19] A. Hertig, "Ethereum's Two Ethereums Explained," 2016, http://www.coindesk.com/ethereum-classic-explained-blockchain/

[20] V. Buterin, "Transaction Spam Attack: Next Step," 2016, https://blog.ethereum.org/2016/09/22/transaction-spam-attack-next-steps/

[21] H. Jameson, "Hard Fork No. 4: Spurious Dragon," 2016, https://blog.ethereum.org/2016/11/18/hard-fork-no-4-spurious-dragon/

[22] Hyperledger, "Hyperledger," https://www.hyperledger.org/

[23] R3CEV, "R3CEV," http://www.r3cev.com/

[24] E. Ben-Sasson, "Zerocash: Decentralized Anonymous Payments from Bitcoin," 2014.