정보보호개론(CSE441)
www.facebook.com/KoreaTech.Crypto

블록체인 2.0

NOTE 16

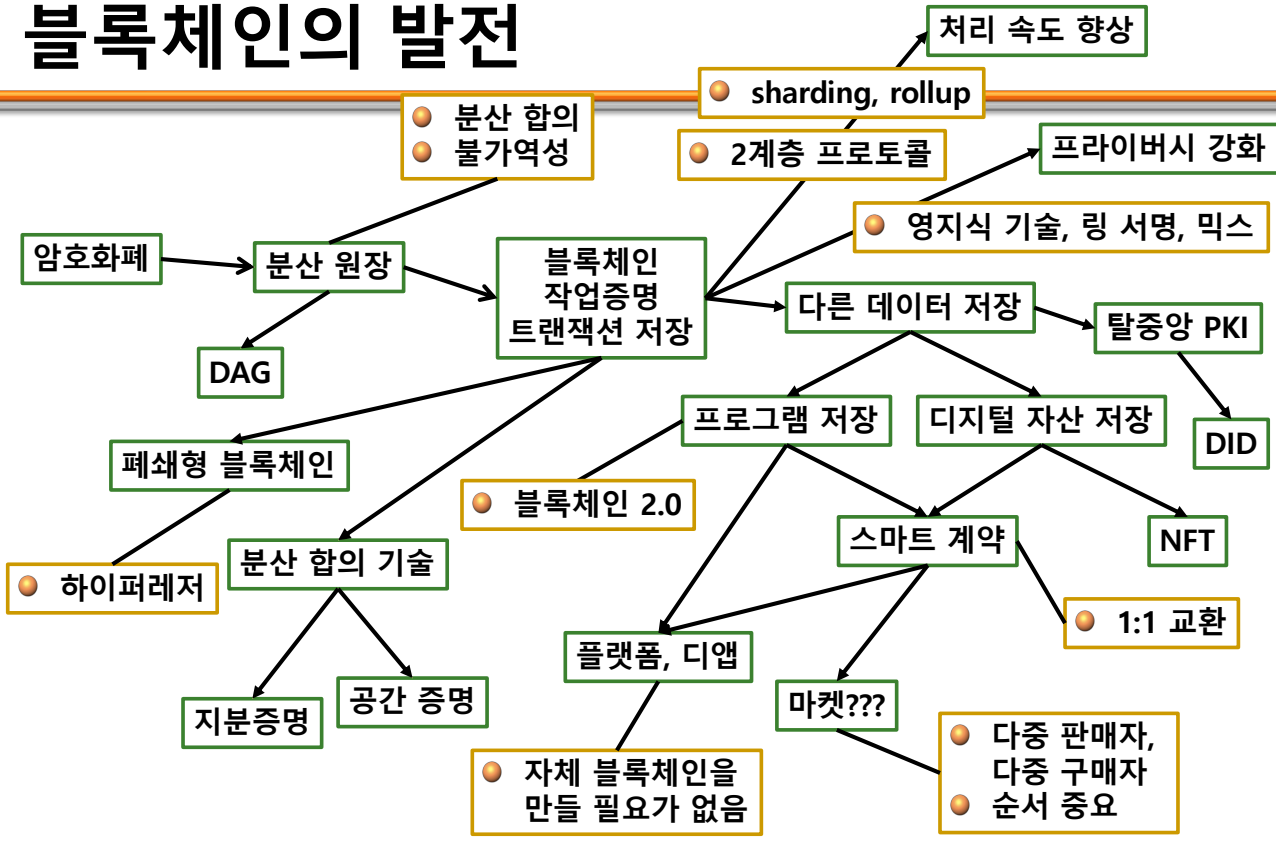
한국기술교육대학교 컴퓨터공학부 김상진

sangjin@koreatech.ac.kr
www.facebook.com/sangjin.kim.koreatech

교육목표

- 블록체인 2.0
  - 이더리움, 이더리움2.0
  - 알트코인
- 분산합의 기술
  - 지분증명
  - 공간증명
  - Tangle
  - Algorand
  - Hedera
- 암호화폐와 블록체인의 미래
  - 폐쇄형, 공개형 블록체인
  - 처리 속도 향상 기술, 익명성 향상 기술
  - Stable coin, NFT, DID

블록체인의 발전



블록체인 2.0

- 일반 데이터 대신에 프로그래밍 코드를 블록체인에 기록하자
  - 무슨 차이? (programmable money)
    - 특정 조건이 충족되면 자동 실행 가능 ⟹ 스마트 계약
    - 프로그래밍을 통해 할 수 있는 것 자체가 무궁무진 (튜링 완전)
  - 장점
    - 실행될 것이 모호하지 않음
    - 무결성 측면에서 안전함
- 이 개념은 비탈릭 부테린이 제안한 이더리움(ethereum) 암호화폐가 최초
  - 백서: 2013
  - 2015년부터 서비스 시작

비탈릭 부테린
https://blog.ethereum.org/author/vitalik-buterin/

이더리움

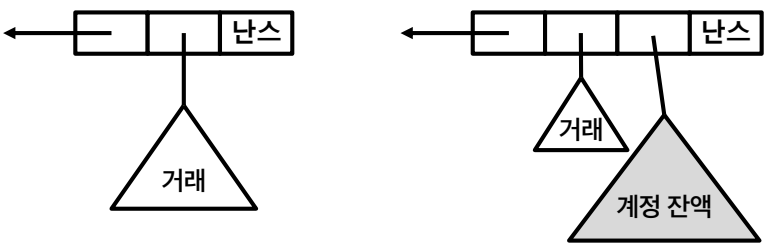
- 단위: 이더(ETH)
  - BTC와 달리 일반 유통을 위한 코인이 아니라 이더리움에서 동작하는 탈중앙화 앱의 사용료로 제한된 코인

2019. 5	2020. 5	2021. 5. 16	2022. 5. 7	2023. 5. 19
205,100원	244,700원	4,627,000원	3,502,000원	2,424,000원

- PoW 방식에서 단계적으로 Casper라는 PoS로 현재 전환이 진행 중임
  - 최초 7,200만개 배포, 각 블록마다 5개 발행
  - 15초마다 새 블록 생성 (비트코인 10분)
- 튜링 완전한 Solidity와 같은 프로그래밍 언어로 작성된 코드를 블록체인에 등록할 수 있음
  - 프로그래밍 코드 자체는 모호하지 않고, 블록체인에 등록함으로써 무결성까지 보장할 수 있음
  - 이 소스는 각 참여자의 컴퓨터에서 실행 가능
    - 자바와 유사한 개념인 EVM(Ethereum Virtual Machine)을 사용

UTXO Model vs. Account Model (1/2)

- UTXO 기반 vs. 계정(account/balance model) 기반
  - 비트 코인은 UTXO 기반, 이더리움은 계정 기반
    - UTXO 기반: 사용자 지갑이 특정 주소의 UTXO가 어느 블록에 저장되어 있는지 유지함
    - 계정 기반: 광역적으로 모든 계정의 잔액 정보를 유지함
  - 확장성, 프라이버시 vs. 단순성, 효율성
    - UTXO 기반: 한 계정의 여러 트랜잭션을 병행 처리할 수 있음
    - 계정 기반: 카운터를 이용하여 한 계정의 트랜잭션을 순차적으로 처리함



UTXO Model vs. Account Model (2/2)

UTXO 기반	계정 기반
	• 기존 은행 모델과 같음 (더 직관적인 모델)
• 트랜잭션 내용 복잡	• 트랜잭션 내용 단순 및 크기가 작음 • 재전송 문제 있음
	• 잔액만 유지하기 때문에 공간 절약
• 지갑이 주소 관련 UTXO를 관리해야 함	• 모든 노드가 모든 계정의 잔액 상태 정보를 유지해야 함
• 병행으로 처리 및 검증 가능함	• 한 계정의 거래는 순차적으로 처리해야 함
	• 프로그래밍하기 용이함
• 여러 주소를 사용하여 강한 프라이버시 제공 가능	• 여러 계정을 사용할 수 있지만 보통 같은 계정을 계속 사용하도록 유도함

UTXO 기반	계정 기반
• 모든 입력은 유효(아직 지불에 사용하지 않아야 함)해야 함 • 모든 입력 소유자의 유효한 서명이 있어야 함	• 한 계정에서 다른 계정으로 금액 이동
• 입력의 총액이 출력의 총액보다 커야 함	• 트랜잭션의 전송 계정이 충분한 금액이 있으면 유효한 지불 (수수료 포함) • 거스름 개념이 없음

Gas Limit, Gas Price

- 1 wei: 0.1^{18} ETH
- 1 gwei: = 10^9 wei = 0.1^9 ETH

- 처리 비용(수수료): 가스 상한(gas limit)과 가스 비(gas price)를 통해 제시함
  - EVM에서 코드 실행 비용
    - 이더리움의 트랜잭션은 코드임
    - 기본 연산(덧셈, 곱셈 등) 수행에 필요한 가스가 결정되어 있음
    - 특정 코드를 실행할 때 필요한 가스는 예측할 수 있음
  - 실제 실행하였을 때 소요된 가스는 gas limit보다 적어야 함
    - 초과하면 취소되지만 수수료는 그대로 지급됨
  - 이 때문에 수수료는 상한이 아니라 가스 비를 조절해야 함
- 높은 비용을 제시한 트랜잭션이 더 빠르게 처리될 가능성이 높음
- 트랜잭션의 실행 주체는 채굴자
  - 블록 체인에 기록되어야 유효한 트랜잭션이 되는 것임
  - 한 블록에 포함할 수 있는 총 가스는 제한되어 있음
  - 수수료는 채굴자가 가지게 됨

- 비트코인은 입력과 출력의 차이로 수수료 계산
- 가스비의 조절로 일정 수준의 수수료 유지 가능
- 주유값과 유사: 1L 금액은 고정되어 있지 않음
- 매일 변할 수 있으며, 주유소마다 다를 수 있음

이더리움의 트랜잭션

- 구성요소
  - 카운터
  - 수수료 정보: 가스 상한, 가스 비
  - 대상 계정 정보: EOA 계정 또는 스마트 계약 계정
  - 금액
  - 데이터: 스마트 계약 실행을 위한 정보 (함수 호출)
  - 전자서명 값 (ECDSA)
    - Sender 정보 포함
- 트랜잭션의 종류
  - 일반 트랜잭션: EOA 간 ETH 교환
  - 스마트 계약 등록 트랜잭션
  - 스마트 계약 실행 트랜잭션

이더리움 채굴과 작업 증명 (1/2)

- 채굴 단계
  - 새 트랜잭션을 수신하면 mempool에 저장함
  - 새 블록을 채굴할 시점이 되면 블록에 추가할 트랜잭션을 선별함
    - 트랜잭션의 유효성 검증
    - 유효한 트랜잭션을 실행하고 트랜잭션 수수료 계산
  - 작업 증명 시작
- 비트코인과 차이점
  - 비트코인은 블록 크기가 제한되지만 이더리움은 블록의 총 가스비로 제한함
    - 가스비의 제한이 크기 제한 역할을 함

- 포함할 트랜잭션을 선택하는 기준은 채굴자 마음
- 가스비가 높은 것을 보통 선호함
- 목표는 수수료를 극대화하는 것이지만 작업 증명을 빠르게 시작하는 것도 필요함

이더리움 채굴과 작업 증명 (2/2)

- 비트코인의 작업 증명의 문제점을 다소 보완하고자 ASIC을 이용한 채굴이 경제성이 없는 알고리즘을 사용함
  - Ethash라는 알고리즘을 사용하며, GPU를 이용한 채굴만 하고 있음
- 이더리움은 15초마다 새 블록 생성
  - 임시 포크가 더 빈번하게 발생할 수 있음
    - 유효한 블록이지만 체인에 포함하지 못한 블록을 uncle(ommer) 블록이라 함
  - 가장 긴 체인 원칙을 사용하지만 삼촌 블록에 대해서도 보상함
    - 채굴 동인 증가, 채굴 집중화 해소, 안전성 증가
  - 채굴자는 이와 같은 블록을 새 블록을 생성할 때 추가할 수 있음
    - 깊이가 최대 6블록 이내, 최대 2개 포함 가능
    - 기존보다 7/8(=4.375 ETH) 보상
- 현재는 Casper라는 지분 증명 방식으로 전환되고 있음 (이더리움 2.0)

스마트 계약

- 스마트 계약(smart contract)
  - 1996년에 Nick Szabo가 처음 소개한 개념
  - 프로그래밍을 통해 스마트 계약을 작성하여 블록체인에 등록할 수 있음
  - 이전 계약과 달리 법률가 대신에 프로그래머가 계약을 작성함
  - 내용이 모호하지 않고 계약이 무조건 이행됨
  - 예) 디지털 상품의 거래
    - 지불자는 상품의 금액만큼의 이더를 계약에 예치함
    - 디지털 상품을 예치함
    - 조건이 만족되면 자동으로 각자에게 전달됨
  - 단점. 계약이 등록되면 취소할 수 없으며, 오류가 포함되면 심각한 결과를 초래할 수 있음. 법적 보호를 받을 수 없음

- 원격에 있는 두 사용자가 내일 축구 경기에 대해 내기를 하고자 한다.
- 스마트 계약은 두 사용자가 안전하게 이 내기를 할 수 있게 해줌

스마트 계약의 장점

- 자율성(autonomy): 중재자 없이 자동 실행됨
  - 계약을 강제화하기 위한 다른 추가적인 장치가 필요 없음
- 신뢰(trust): 계약이 투명하고 자동적으로 실행되며, 안전하기 때문에 믿을 수 있음
- 백업(backup): 자동으로 중복 저장됨 (P2P 블록체인)
- 안전성(safety): 계약의 조작 가능성이 계산적으로 힘듦
- 신속성(speed): 소프트웨어를 통해 자동 실행
- 경제성(savings): 중재자가 필요 없어 경제적임
- 정확성(accurracy): 소프트웨어로 처리되기 때문에 사람 실수에 의한 오류는 발생하지 않음


```
pragma solidity >=0.4.0 <0.6.0;

contract SimpleStorage{
    uint storedData;
    function set(uint x) public {
        storedData = x;
    }
    function get() public view return (uint) {
        return storedData;
    }
}
```

```
/* a contract attempts to get the coins */
function transferFrom(address _from, address _to, uint256 _value) returns (bool success) {
    if (balanceOf[_from] < valuie) throw;                 // check if the sender has enough
    if (balanceOf[_to] + value < balanceOf[_to]) throw; // check for overflow
    if (_value > allowance[_from][msg.sender]) throw; // check allowance
    balanceOf[_from] -= value;
    balanceOf[_to] += _value;
    allowance[_from][msg.sender] -= _value;
    Transfer(_from, _to, _value);
    return true;
}
```

계정의 종류

- EOA(Externally Owned Account)
  - 비트코인의 계정과 유사 (개인키, 공개키 쌍으로 제어)
  - 이더 잔액 유지
  - 다른 EOA로 ETH를 전송할 수 있음
  - CA로 트랜잭션을 보내 코드를 활성화할 수 있음
  - 스마트 계약을 저장할 수 없음
- CA(Contract Account): 단순 계약 계정, 다중 서명 계약 계정
  - 이더 잔액 유지
  - 스마트 계약이 저장되어 있음
  - 스스로 트랜잭션을 만들 수 없음
  - 다른 EOA나 CA로부터 트랜잭션(또는 메시지)을 받으면 계약이 실행될 수 있음
    - 스스로 활성화되지 않음
    - 수행하기 위해 비용 지불이 필요함

DApp (Decentralized Application)

- 특징
  - 오픈소스: 신뢰(↑), 안전성(소스 자체가 블록체인, 수정문제: ↑↓)
    - 여러 스마트 계약을 통해 구현 및 동작함. 소스 갱신이 어려울 수 있음
  - 내부 통화 사용: 토큰 형태로 사용료 지급, 유료 서비스
    - 디앱 사용을 위해 해당 플랫폼 지갑 및 코인 보유 필요
    - 사용자에게 이 토큰으로 인센티브 지급 가능
  - P2P 네트워크에서 동작 ⟹ 단일 실패점이 없음 (토렌트와 유사)
  - 앱을 통제하는 단일 주체가 없음
- 모바일앱과 비교
  - 모바일앱: 중앙 서버(데이터베이스 포함)와 연계된 서비스
  - DApp: 분산된 블록체인을 활용
    - 하지만 앱의 모든 데이터를 블록체인에 기록하는 것은 적절한가?
  - 가용성이 우수함: 단일 실패점이 없음
  - Front end는 둘 다 유사

토큰과 ICO

- 이더리움 토큰과 ICO(Initial Coin Offering)
  - 이더리움은 디앱에서 사용할 수 있는 토큰을 쉽게 발행하고 사용할 수 있도록 함
  - 이더리움 토큰은 이더로 교환할 수 있고, 이더는 명목화폐로 바꿀 수 있음
  - ICO는 이더리움 기반 디앱을 만들거나 만들 기획(백서 작성)을 하고, 이 앱에서 사용할 토큰을 사전에 판매하여 자금을 조달하는 것을 말함
  - ICO를 하는 기업은 실제 상장한 후에 앱 개발에 성공하지 못할 수 있으며, 일부로 개발하지 않을 수 있음

Ethereum 2.0 (Serenity) (1/2)

- 2023년에 완전 전환을 목표하고 있음
- 샤딩 도입: 블록체인의 수평 분할 (다음 슬라이드 + 36)
  - 1초 30개 트랜잭션 처리 ⟹ 1초 100,000개 트랜잭션 처리
- Beacon chain
  - PoS와 샤딩 도입을 위한 체인으로 메인 체인과 별도 운영
    - 최종적으로 메인 체인과 결합하여 하나의 체인으로 동작할 예정
  - 정해진 시간 주기(epoch, slot)로 동작함
    - epoch: 32개 slot으로 구성
    - slot: 12초마다 하나의 비콘 블록과 64개 샤드 블록이 추가됨
    - 병렬로 운영되는 샤드 체인을 연결하는 역할
  - 활성 검증자(validator) 집합 관리 (PoS로 전환)
    - 32ETH를 예치해야 검증자로 참여할 수 있고, 활동하지 않으면 잃게 됨
  - 제안자 선정에 사용하는 의사난수 process RANDAO 제공
  - 블록 확정, cross shard 트랜잭션 처리

Ethereum 2.0 (Serenity) (2/2)

- PoS 도입
  - 검증자 중 한 명을 랜덤하게 결정함. 이를 제안자(proposer)라 함
    - 제안자는 블록을 제안할 때 원하는 금액의 ETH를 예치함
    - 예치액에 따라 블록에 포함할 수 있는 트랜잭션 수가 결정됨
    - 제안자는 블록 보상과 수수료를 받음
    - 유효하지 않은 트랜잭션을 블록에 포함하면 예치 금액을 잃게 됨 (slashing)
  - 제안자로 선정되지 않은 검증자들은 입회자(attester) 역할을 함
    - 제안자가 제안한 블록을 검증(up/down 투표)함
    - 다른 검증자의 부정행위를 감시함
    - 2/3 이상의 up 투표가 되면 비콘 체인에 포함되어 확정됨
  - 한 slot마다 하나의 제안자와 최소 128명의 입회자로 구성된 위원회 (committee)가 참여함

알트코인 (1/2)

- 비트코인 이후 등장한 유사한 형태의 암호화폐를 말함
- 주요 알트코인
  - Namecoin: key/value 등록. 기존 비트코인은 트랜잭션만 블록체인에 기록된다는 단점을 극복하기 위해 고안됨
  - Peercoin: 최초 지분증명(PoS)을 사용하는 암호화폐
  - Ripple: 글로벌 송금을 목적, 단위는 XRP
    - 변형된 비자틴 합의 프로토콜을 사용함
  - Ethereum Classic: 최초 이더리움. DAO 공격 문제를 처리하기 위해 부타린은 하드 포크를 하였으며, 이것에 동의하지 못한 일부는 기존 이더리움을 계속 채굴하여 만들어진 암호화폐
  - Zcash: 영지식 증명을 이용하여 비트코인보다 코인 소유자의 프라이버시를 강화한 암호화폐

알트코인 (2/2)

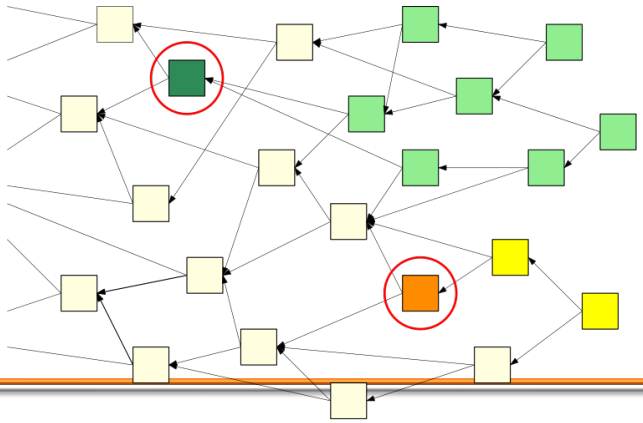
- IOTA: 사물인터넷을 목표
  - Tangle이라는 새로운 블록체인 기술 사용
- EOS: Ethereum을 개선한 암호화폐. EOS를 블록체인 3.0이라고도 함
  - 이더리움과 달리 PoS를 사용하며, 분산 앱 개발에 필요한 다양한 리소스를 제공함
- ENJIN: 게임을 위한 암호화폐
- Algorand: PPoS(Pure PoS) 방식의 분산합의 사용
- Monero: 링 서명 활용
- 이 외에도 Litecoin, Dogecoin, Dash, Stellar, Cardano, Tron, FileCoin 등 무수히 많은 알트코인이 등장하고 있음

DAG 기반 분산 원장

- IOTA
- Nano

IOTA

- IoT를 위한 암호화폐
- 탱글(tangle)이라는 새로운 형태의 분산 합의 알고리즘을 사용함
  - 작업 증명도 아니고 지분 증명도 아님
    - 채굴자도 없고, 검증자를 선정하지 않음
  - 트랜잭션을 DAG를 구성함
  - 트랜잭션을 생성하는 사용자는 검증자 역할까지 해야 함
  - 블록 단위로 트랜잭션을 모아 확정하지 않음
  - 수수료가 없음
- 화폐는 최초 트랜잭션에서 모두 발행됨



Tangle

- 새 트랜잭션은 항상 2개의 기존 트랜잭션과 연결되고, 트랜잭션들은 DAG를 형성함
  - 각 사용자는 새 트랜잭션을 그래프에 포함하기 위해 기존 2개의 트랜잭션(tip)을 승인(검증)해야 함
    - Tip: 다른 트랜잭션에 의해 승인되지 않은 트랜잭션
  - Tip 선택 알고리즘 (https://vo.la/T99e)
    - 최초 트랜잭션부터 그것을 승인한 트랜잭션을 따라 tip을 만날 때까지 그래프를 탐색함
    - 트랜잭션을 승인한 트랜잭션 중 임의로 선택하거나 weight가 높은 트랜잭션을 선택하여 탐색을 진행함
      - 트랜잭션의 weight: 직간접적으로 승인한 트랜잭션 수
- 트랜잭션이 많은 트랜잭션들에 의해 직간접적으로 승인되면 확정됨
  - 바로 확정되지 않음
- 보상이 없지만 정해진 방식대로 하지 않으면 자신의 트랜잭션이 승인되지 않을 수 있어 정해진 방식을 사용하도록 유도하는 방식

Nano (1/3)

- 원래 RaiBlock 이름으로 시작한 암호화폐
- 수수료도 없고, 특별한 보상체계를 사용하지 않음
- 계정마다 별도 체인을 유지함
  - 블록 단위가 아니라 트랜잭션 단위 체인
    - 각 계정의 체인은 계정 소유주만 갱신할 수 있음
      - 포크의 발생은 프로그래밍 오류 또는 계정 소유주의 부정 행위
  - 각 계정은 다른 한 계정을 대리인으로 지정함
    - 본인이 온라인 상태가 아닐 때 투표를 대신해주는 역할
    - 대리인 수행에 대한 보상은 없음
  - 계정 별 독립 체인이므로 트랜잭션에 의해 영향을 받는 계정은 항상 2개임
    - 다른 계정은 독립적으로 상호 독립적으로 진행할 수 있음

Nano (2/3)

- 한 트랜잭션은 출금과 입금 트랜잭션으로 구성되어, 계정 A 체인에는 출금, 계정 B 체인에는 입금 트랜잭션이 추가됨
  - B가 입금 트랜잭션을 추가해 주어야 완료되는 형태
    - 트랜잭션의 상태: 완료(settled, 입금 트랜잭션이 수취인 체인에 추가된 상태), 진행중(unsettled, 불일치 문제 존재)
  - 출금 트랜잭션이 전송되면 그것을 취소할 수 없음
  - 각 입금, 출금 트랜잭션은 블록 소유주의 개인키로 서명됨
  - 출금과 입금 간 연결되므로 전체 계정은 격자 형태의 DAG를 형성함
  - 입금 트랜잭션의 추가 순서는 소유주 마음
  - 각 계정 체인의 트랜잭션은 현재 계정 잔액 정보를 유지하고 있으므로 다른 노드들은 다른 계정의 최종 트랜잭션만 유지할 수 있음
    - Aging 기능 제공

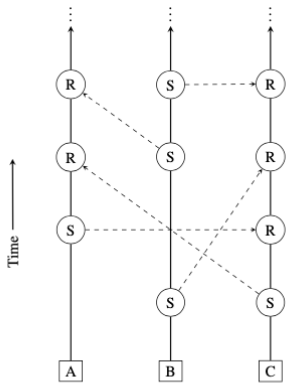


Fig. 3. Visualization of the block-lattice. Every transfer of funds requires a send block (S) and a receive block (R), each signed by their account-chain's owner (A,B,C)

Nano (3/3)

- 각 트랜잭션은 PoW를 해결해야 함
  - 해시 퍼즐은 지난 트랜잭션과 난스를 이용하여 계산
    - 트랜잭션 전송 후 바로 다음 트랜잭션을 위한 PoW를 수행할 수 있음
  - 스팸 방지용: 한 계정이 너무 많은 트랜잭션을 생성하여 네트워크 운영에 영향을 주는 것을 방지하기 위한 용도
- 포크의 발생
  - 4라운드의 투표 진행 (모두가 참여)
    - ORV(Open Representive Voting)
  - 각 투표자의 투표 가중치는 그 투표자가 대리하는 계정의 잔액 합
  - 포크의 발생은 해당 계정에만 영향을 주며 다른 계정의 거래 진행에 전혀 영향을 주지 못함

작업증명 외 분산 합의 기술

- 지분 증명
- 공간 증명
- 기타
  - 순서에 대한 합의

지분 증명 (1/2)

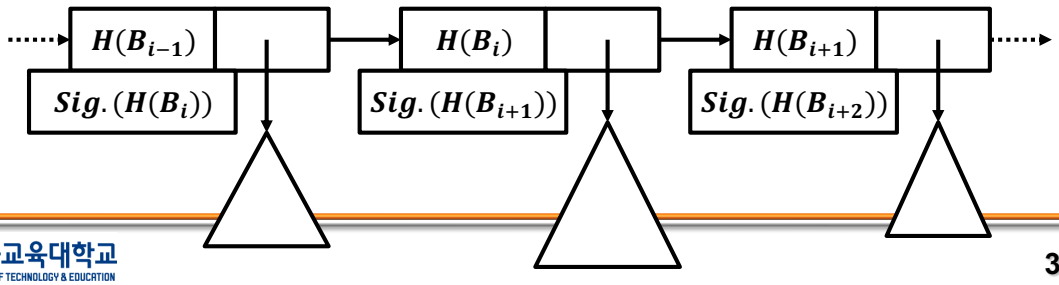
- 블록을 생성할 수 있는 검증자를 결정적 방법으로 결정함
  - 블록 생성을 통해 화폐를 발행하지 않음
  - 작업 증명처럼 경쟁을 하지 않음 (전기 소모가 발생하지 않음)
- 결정하는 방법에 따라 다양한 PoS가 가능
  - 보통 확률과 부(소유한 코인양)에 의해 결정
  - 예) Peercoin: 지갑의 나이와 보유 코인 수로 결정
    - n일 동안 이 지갑을 사용하지 않으면 이 지갑의 나이가 n임
    - 검증자가 되면 새 코인을 받기 때문에 한번 검증자가 되면 최소 몇 일이 지나야 다시 검증자가 될 수 있음
- 종류: 체인 기반, BFT 기반
  - 체인 기반은 단일 검증자, BFT는 다중 검증자 사용
  - BFT 기반: 다중 검증자 간의 비잔틴 합의 수행 (투표 진행)

- 검증자 역할 수행 동인: 수수료 수입

- 51%의 지분 소유
  - 51%의 해시 파워를 소유하는 것보다 더 힘듦

지분 증명 (2/2)

- 검증자의 부정을 막기 위해 검증자 역할을 하기 위해 일정 금액을 예치하는 방식도 사용함
  - 예) 이더리움의 Casper
- 각 블록은 이전 블록 값과 함께 검증자에 의해 전자서명됨
  - 여전히 체인 형태임
- 지분 증명의 안전성
  - 검증자가 부정하지 않는다면 신뢰 기관을 이용한 중앙집중식과 동일함
    - 제3자들은 블록체인의 내용을 수정, 삭제할 수 없음
  - 검증자가 할 수 있는 공격도 제한적임
- PoW와 PoS를 혼합하여 사용할 수 있음



Algorand (1/2)

- MIT의 S. Micali 교수 중심으로 제안됨
- UTXO 기반: 각 사용자는 전자서명하여 트랜잭션 생성
- PoS 형태의 분산 합의 사용 → PPoS(Pure PoS)
- r번째 블록: $B^r = (r, PAY^r, Q^r, H(B^{r-1}))$
  - PAY^r: 블록에 포함되는 트랜잭션 집합
- 암호기술을 이용하여 리더와 검증자들을 선출함
  - 리더가 블록을 생성하고 검증자들은 BFT를 통해 블록을 확정함
    - 전체 사용자의 일부가 각 라운드마다 검증위원회가 됨
  - 사전에 누가 리더가 될지 검증자가 될지 알 수 없음
    - 각 개인은 확인할 수 있음
    - 나중에 자신이 정당한 리더이고 검증자임을 증명할 수 있음
- 포크가 발생하지 않음
  - 과반수가 넘는 서명을 받은 블록이 여러 개 존재할 수 없음

Algorand (2/2)

- 스스로 이전 블록의 Q 값을 이용하여 리더 또는 검증자 당선 여부를 확인 가능
- 누가 리더/검증자로 당선될지 예측할 수 없음

- 합의 알고리즘
  - 단계 1. 리더 선출
    - 각 사용자 i는 $\sigma_i = SIG_i(r, 1, Q^{r-1})$를 계산하고 . $H(\sigma_i)$를 계산함 (σ_i는 각 i만 계산 가능)
    - 이 값이 정해진 p이하 이면 리더가 될 수 있음
    - 리더 역할을 하기로 한 사용자는 다음 블록을 준비하여 전파함:
      $B^r = (r, PAY^r, SIG_i(Q^{r-1}), H(B^{r-1})), SIG_i(B^r), \sigma_i$
  - 단계 2. 검증자 선출
    - 각 사용자 i는 . $H(SIG_i(r, s, Q^{r-1}))$를 계산함
    - 이 값이 정해진 p'이하 이면 검증자가 될 수 있음
    - 검증자는 자신이 수신한 다음 블록 중 . $H(\sigma_i)$가 가장 작고, 유효한 블록을 보낸 사용자를 리더로 결정하고 다른 검증자와 BFT를 수행하여 확정
      - BFT 이후 다수 검증자의 서명이 블록에 추가됨

공간 증명(proof of space) (1/5)

- 공간 증명은 일정한 크기의 디스크 공간을 소모(waste)하였음을 증명하는 것임
  - 작업 증명(PoW)은 일정 연산을 수행하였음을 증명
  - 가지고 있는 디스크 공간에 랜덤한 데이터를 저장해야 하며, 정기적으로 이 데이터를 계속 해당 공간에 유지하고 있다는 것을 증명해야 함
    - 이 공간을 다른 유용한 용도로 활용하지 못함
  - 작업 증명과 달리 에너지가 소모되는 것은 아니며, 특수 하드웨어를 통해 속도를 향상할 수 없음
- ID 발급에 적용 예)
  - ID를 발급할 때마다 일정한 디스크 공간을 소모하도록 함
    - Why? 악의적인 사용자가 많은 수의 ID를 만드는 것을 방지함
  - 최초 만들 때 공간 증명을 한번만 하면 의미가 없음
    - 사용할 때마다 또는 주기적(예, 하루에 한 번)으로

- 서버는 공간에 저장할 랜덤 데이터와 그 데이터의 블록 단위 MAC 값을 전달함
- 사용자는 이를 유지한 후에 서버의 요청이 있을 때마다 POR를 수행함

공간 증명 (2/5)

- 두 가지 단계로 구성
  - 초기 단계:
    - 증명자는 N개 블록으로 구성된 데이터를 저장해야 함
  - 증명 단계: 초기 단계 이후 주기적으로 계속 수행해야 함
    - 증명자는 저장된 공간에 기대하는 데이터가 있는지 증명함
      - 확인자는 N개 중 랜덤한 블록을 요청하여 이들을 확인함
  - ID 발급 예) 계정 개설: 초기단계+증명 단계, 주기적: 증명 단계
- 효율성 요구사항
  - 모든 단계가 효율적이어야 함
  - 하지만 초기 단계는 저장하는 블록 수 N에 비례할 수밖에 없음
    - 초기 단계는 한번만 수행하는 것임
- 안전성 요구사항
  - 증명자는 N개 블록을 실제 저장하여 유지하지 않고 있으면 성공적으로 증명할 수 없어야 함

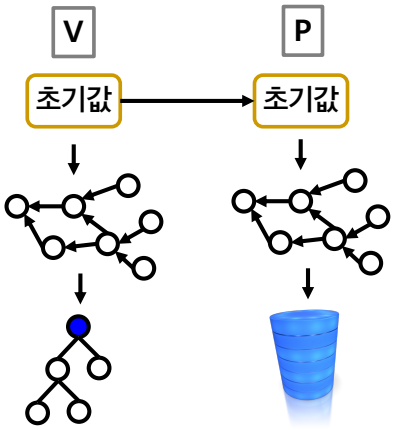
- POR에서 증명자는 서버, 확인자는 사용자

공간 증명 (3/5)

- 초기 단계를 어떻게 진행?
  - 시도 1. 초기 단계: V ⟹ P: N크기의 랜덤 데이터 전달
    - 전달해야 하는 데이터 양이 많아 요구사항을 충족하지 못함
    - P는 이 데이터를 저장하지 않고 데이터를 압축하여 저장할 수 있음
    - V는 이 랜덤 데이터를 의사난수 함수를 이용하여 생성하였다면 이때 사용한 랜덤 seed만 유지 가능
      - POR에서 확인자는 실제 데이터를 유지할 필요는 없음
  - 시도 2. 초기 단계: V ⟹ P: seed 전달
    - P가 seed만 유지할 수 있음

공간 증명 (4/5)

- Dziembowski 등이 제안한 작업 증명에 사용한 기술
  - Pebble 그래프: N개의 노드로 구성된 DAG(주기가 없는 방향 그래프)
    - 초기 확인자가 제시한 입력을 이용하여 그래프를 생성함
    - 하지만 생성한 그래프의 각 값을 저장하지 않거나 일부만 저장하면 증명 수행이 가능하지 않음
      - 확인자가 전달한 입력만 유지하면 다시 그래프를 생성해야 함



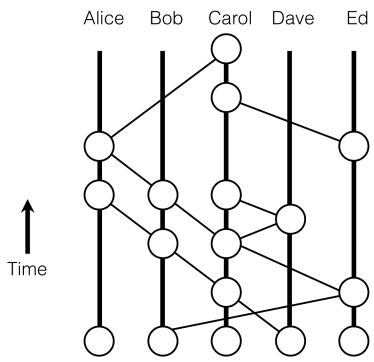
  - Merkle 트리
    - 증명자는 pebble 그래프를 생성한 후에 각 노드 값을 이용하여 Merkle 트리를 만들어 그것의 루트 값을 확인자에게 전달해야 함
    - 증명 단계에서 확인도 이 루트를 활용함

공간 증명 (5/5)

- 공간 증명으로 블록체인을 어떻게
  - 상호작용 공간 증명을 비상호작용 증명으로 바꿈
  - 모든 채굴자들은 비트코인처럼 거래들을 모아 그것의 머클 트리, 이전 공간 증명 값을 이용하여 공간 증명을 수행함
  - 공간 증명 결과 값에 따라 승자를 결정하는 규칙을 사용함
- 공간 증명을 암호화폐에 적용한 예
  - spacecoin
  - chia
  - filecoin

Hedera

- 순서에 대한 분산 합의
  - 트랜잭션을 전달할 때 내가 최신에 받은 트랜잭션, 내가 최신에 보낸 트랜잭션을 연결하여 전송하면 모든 참여자가 동일한 해시 그래프를 그릴 수 있고, 동일한 해시 그래프를 그릴 수 있으면 순서에 대해 분산 합의를 할 수 있음
  - 합의된 순서는 모든 사용자가 수신한 시간의 평균
  - 기존?
    - 블록체인: 채굴자 마음



Private vs. Public 블록체인

- 분류 기준
  - 블록체인의 블록 값을 누가 생성할 수 있는지에 따라 구분
  - 블록체인 서비스에 참여를 개방하는지 제한하는지에 따라 구분
  - 블록체인에 기록된 데이터를 누가 볼 수 있는지에 따라 구분
- 공개형 블록체인: 비트코인, 이더리움 등
- 보통 기업에서 내부 목적으로 운영하는 블록체인은 폐쇄형임
  - 폐쇄형도 일반 블록체인의 특성(append-only, immutable, distributed)을 충족해야 함
  - 폐쇄형에서도 신뢰구축이 가능한가?
    - 기관이 복수의 신뢰서버를 구축하고 이 서버들이 전자서명과 해시함수를 이용하여 체인을 만들 경우 문제가 없나?
      - 신뢰서버가 해킹되지 않는 이상, 또는 신뢰서버를 특정 개인이 마음대로 수정이 가능하지 않는 이상 해당 응용에 문제가 되지 않을 수 있음

Permissioned, Consortium Blockchains

HyperLedger

- 폐쇄형 >> 더 정확하게는 consortium 블록체인
  - 2015년 리눅스 재단에 의해 시작된 오픈 소스 블록체인 프로젝트
    - IBM, SAP, Intel 등 다양한 기업이 참여
    - 예) Fabric: IBM이 주도
- Fabric 동작원리
  - 단계 1. 응용은 트랜잭션을 생성하고 피어들에게 요청
  - 단계 2. 피어들은 트랜잭션을 직접 실행 후 결괏값에 서명하여 되돌려 줌
  - 단계 3. 응용은 단계 2에서 받은 정보를 정렬자에게 전달 함
  - 단계 4. 정렬자는 단계 3에서 받은 정보를 확인하고 승인함
    - 트랜잭션의 순서를 결정하는 역할도 함
  - 단계 5. 피어들은 승인된 내용을 원장에 기록함

처리 속도 향상 기술

- 샤딩(sharding)
  - 데이터베이스를 분할(partitioning)하여 여러 서버에서 서비스를 제공하는 기술 (게임 산업에서 부하 분산을 위해 많이 사용)
  - 하나의 블록체인에 모든 트랜잭션을 유지하는 것이 아니라 일정한 기준으로 나누어 병행 처리하는 기술
    - 나누어진 블록체인을 샤드(shard)라 함
  - 노드에 샤드를 할당하는 문제, 다른 샤드 간 거래 문제, PoW 방식의 합의 기술을 사용할 경우 해시 파워의 분산으로 개별 샤드가 공격에 취약할 수 있는 문제 등의 해결이 필요함
  - 개별 샤드의 취약점 문제
    - PoS에서는 검증자의 랜덤 shuffling

처리 속도 향상 기술

- ZK Rollup, Optimistic Rollup
  - 여러 트랜잭션을 묶어 처리하는 기술
  - ZK 롤업은 영지식 기술을 이용하여 여러 유효한 트랜잭션이 수행되어 이전 상태에서 현재 상태로 올바르게 상태 전환이 되었다는 증명을 제시하는 방법
  - 난관적 방법은 검사하지 않고 진행하지만 부정행위를 할 경우 벌을 줄 수 있도록 하는 방법

익명성 향상 기술: Monero (1/2)

- 링 서명: 여러 개의 UTXO 중 어느 것이 실제 sender의 UTXO인지 알 수 없음
  - 링 크기: 11 (처음에 유동, 현재는 링 크기 정보에 의한 노출 때문에 고정)
  - 이중 지불 문제를 위해 트랜잭션에는 key image가 포함됨
    - Key image: $I = xH_p(P)$
      - x: sender 서명키, $P(= xG)$: UTXO의 stealth 주소
    - Sender는 key image의 유효성에 대한 증명을 링 서명에 포함
    - 이 값의 중복 여부를 통해 이중 지불 여부를 검사함
- Stealth address: 수신자 주소를 숨기기 위한 방안 (암호화하여 숨김)
  - $P = H_s(rA)G + B, R = rG \Longrightarrow P$: stealth 주소, R: 트랜잭션 공개키
    - G: 타원곡선 점, r: sender가 선택한 랜덤값,
      $A = aG, B = bG$: 수신자의 공개키,
  - 오직 실제 주소의 개인키를 가지고 있는 경우만 이를 식별할 수 있음
    - $D = H_s(aR) \Longrightarrow DG + B\,?= P$
  - $x = H_s(rA) + b$

익명성 향상 기술 (2/4)

- Pedersen Commitment: 전송 금액을 숨기기 위해 사용
  - $c = rG + aH$, a: commit하는 값, r: 랜덤값
    - G, H: 타원곡선 위의 점
  - 동형(homomorphic) 암호 특성을 가지고 있음
    - 트랜잭션 입력의 합과 출력의 합이 같음을 개별 값을 열지 않고 확인할 수 있음
    - 출력 중 수수료는 공개함
    - 입력: $c_i = r_iG + a_iH$, 출력: $o_i = \gamma_iG + \alpha_iH$, $f = 2H$
      - 수수료: \$2, $\sum r_i = \sum \gamma_i$
        - $f = bG + 2H$, $b = \sum r_i - \sum \gamma_i$ (수수료도 숨길 수 있음)
      - $c = \sum c_i$, $o = f + \sum o_i$, $c - o$이 0과 같은지 확인

- 전송 금액은 트랜잭션을 연결하는 또 다리 고리가 될 수 있음
- UTXO의 출력과 새 트랜잭션의 입력 금액이 같으면...

익명성 향상 기술 (3/4)

- Zcash
  - zk-SNARK 영지식 기술 이용
    - Sender, Receiver, 트랜잭션 금액을 숨기고 거래의 유효성을 증명
  - 증명의 구성
    - 입력 합계와 출력 합계 일치
    - 개인키 소유
    - Nullifier: 이중 사용 방지 요소

익명성 향상 기술 (4/5)

- Dash: 코인 믹스 기술 사용 (Litecoin에서 fork, PoW, 2014)
  - 가지고 있는 코인을 한 곳에 모아 섞은 후 다시 재분배
    - 트랜잭션의 거래 기록 연결 고리를 제거
    - 실제 3개의 코인을 섞음
  - 코인 믹스의 기본 개념인 coin join ⟹ 트랜잭션 join

in	out
5BTC: A	3BTC: B
	2BTC: A

in	out
6BTC: C	3BTC: C
	3BTC: D

coin join

in	out
5BTC: A	3BTC: B
6BTC: C	2BTC: A
	3BTC: C
	3BTC: D

- 트랜잭션을 결합하면 이전과 달리 거래의 당사자를 연결하기 힘듦
- 위 예는 거래 금액 때문에 연결고리가 있지만 금액이 같으면...

익명성 향상 기술 (4/5)

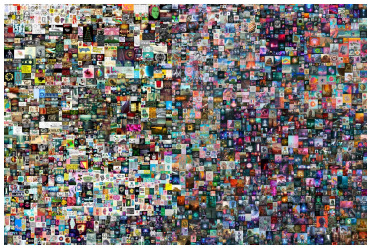
- 재분배하기 위해서는 정해진 액면가로 먼저 분할해야 함
- 재분배는 Masternode가 수행함
  - 참고. Masternode를 통해 즉시 결재 기능도 제공
- 문제점
  - Masternode는 연결고리를 알고 있음
    - 여러 번 섞을 경우(섞을 때 라운드 수 지정, 2~16 라운드, 각 라운드는 다른 masternode가 진행함) 이들이 모두 공모하지 않는 이상 찾기 힘듦
  - Mixing 참여 사실은 기록으로 남음

stable coin

- 암호화폐가 실제 거래에 사용하기 위해서는 암호화폐의 가격 변동성이 적어야 함
  - 이 때문에 개발된 것이 스테이블 코인임
- 스테이블 코인: 기존 화폐 또는 실물 자산(예: 금)과 연동하여 안전성을 보장하는 암호화폐
  - 법정화폐 담보형: 예) Tether(USDT), TrueUSD, Diem(Libra)
    - 기관에 법정 화폐를 담보로 예치
  - 암호자산 담보형: 예) bitShares, MakerDAO
    - 기관에 암호 화폐를 담보로 예치
  - 무담보형: 예) Basecoin, Terra
    - 코인의 유통량 조절하여 코인 가격을 유지하는 방식
- CBDC(Central Bank Digital Currency)는 당연히 스테이블 코인 형태가 될 것임
  - 암호화폐를 이용한 거래의 이점은? 신뢰성, 안전성, 비용절감, 혁신 등

NFT

- NFT(Non-Fungible Token, 대체 불가 토큰)
- 디지털 자산에 대한 소유권 인증서
- 블록체인을 통해 발행, 소유권 변동 내역을 확인하고, 관리할 수 있음
  - 소유권을 나눌 수 있음
- 메타데이터(소유권, 디지털 자산 링크 정보)만 블록체인에 저장함
- 스마트 계약으로 표현 가능
  - 제작한 예술가에게 계속 수익 보장이 가능
- 이더리움 기반 (ERC-71)
- 다른 토큰/코인과 차이점
  - 동등하게 교환할 수 없음. 각 토큰의 가치가 다름
- 문제점
  - 자신의 소유권이 아닌 디지털 자산을 발행할 수 있음
  - 원본 소실 가능



Beeple sold an NFT for $69 million

Through a first-of-its-kind auction at Christie's

암호화폐 활성화가 가져올 변화

- 중개자가 필요한 기존 서비스를 P2P 거래로 전환
  - 부동산, 해외송금 등
    - 문제점.
      - 중개자가 없으면 불편할 수 있음 (서비스 품질)
      - 서비스 플랫폼이 구축되어 있어야 하며, 이를 구축한 기업도 수익 모델이 필요함 (https://averspace.com)
- Prosumer에 대한 보상 방식의 서비스 활성화 가능
  - 예) https://storj.io: 본인의 여분 저장 공간을 제공하고 보상을 받음
  - 예) steemit: 콘텐츠 생성자에 대해 보상함
  - 예) vevue.com: 음식점 동영상을 촬영하여 공유하면 보상함

블록체인의 활용

- 예) 현재 금융 기관들은 블록체인에 대한 연구를 많이 하고 있으며, 향후 대부분의 금융 기관들이 거래 기록들을 블록체인에 저장할 것으로 예측됨
- 예) 에스토니아 정부는 정부의 모든 기록을 블록체인에 저장하고 공개함
  - GuardTime의 KSI(Keyless Signature Infrastructure)
- 예) 블록체인 기반 선거시스템
  - agora.vote, polys.me
- 예) 자동차의 주행기록 저장하여 중고차 시장에 활용
  - https://www.coindesk.com/blockchain-startup-tracks-vehicle-mileage-with-bmw/
- 예) 각종 로그 정보 기록
- 예) 디지털 여권, 각종 증명서 저장 → DID
  - 보통은 증명서 자체가 아니라 증명서에 대한 디지털 증거를 블록체인에 저장
  - 이점. 접근 용이성, 중재자 없이 검증 가능, 무결성 보장, 비용 절감, 공정성 등

  - http://goo.gl/rJ8MQj
  - http://goo.gl/zQSvqx

블록체인과 일반 데이터베이스의 차이

- 블록체인과 마찬가지로 데이터베이스에 데이터를 저장할 때 기록 권한을 가진 자가 전자서명하여 저장하면 서명키가 없는 사용자들은 데이터를 수정하기 어렵게 만들 수 있음
- 블록체인은 기본적으로 데이터가 분산 저장됨
  - 데이터베이스를 중복 유지하여 유사한 효과를 얻을 수 있음
- 그러면 무슨 차이?
  - 블록체인은 체인으로 엮여 있어 첨삭 전용임
  - 기존 데이터를 서명할 때 사용한 서명키를 가진 사용자도 데이터를 수정할 수 없음