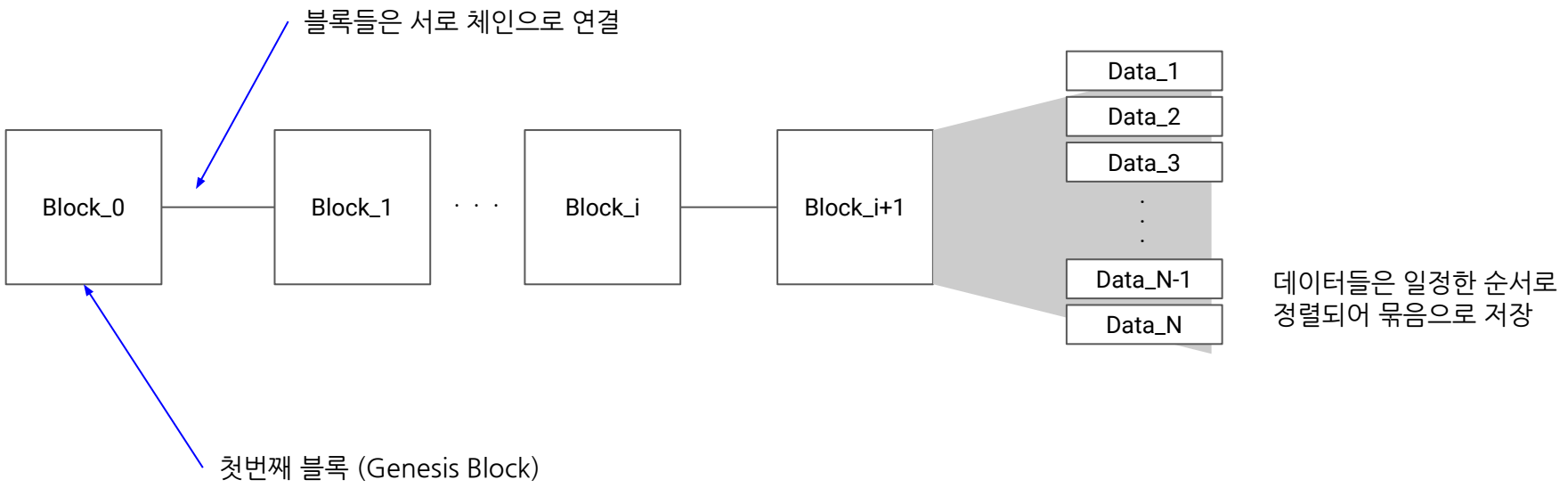
Basics of Blockchain

2019.09.05

블록체인이란?

정보를 **블록**이라고 하는 단위로 저장하여 저장된 블록들을 **체인**형태로 묶은 저장기술



해시 함수 (Hash Function)

임의의 길이의 데이터를 **고정된 길이의 데이터**로 매핑하는 함수

- 해시, 해시 값, 해시 코드 = 해시 함수에 의해 얻어지는 값
- 데이터를 X, 해시함수를 H라고 표기할 때 해시를 H(X)로 표기
- 산업에서 가장 널리 쓰이는 해시 함수는 SHA-2 (e.g., SHA-256), SHA-3 (e.g., Keccak)

Rules

1. 하나의 데이터에서 오직 단 하나의 해시가 도출
2. 임의의 데이터 X와 Y가 있을 때
    a. if X == Y then H(X) == H(Y)
    b. if X != Y then H(X) != H(Y)
    c. if H(X) == H(Y) then X == Y

해시 함수 (Hash Function) 예제 1

같은 함수로 다른 데이터를 해시했을 경우

- 문자열 ‘hello!’를 SHA-256으로 해시한 결과는 다음과 같다:

  `CE06092FB948D9FFAC7D1A376E404B26B7575BCC11EE05A4615FEF4FEC3A308B`

- 문자열 ‘hello?’를 SHA-256으로 해시한 결과는 다음과 같다:

  `B45CF64669F2F8DA6C6CC2DB0329EC1A37D067B9AB7640C029CFD44EB4BF928A`

같은 함수라도 다른 데이터를 해시할 경우 결과값이 크게 다른 것을 확인할 수 있다.

해시 함수 (Hash Function) 예제 2

다른 함수로 같은 데이터를 해시했을 경우

- 문자열 ‘hello!’를 SHA-256으로 해시한 결과는 다음과 같다:

  `CE06092FB948D9FFAC7D1A376E404B26B7575BCC11EE05A4615FEF4FEC3A308B`

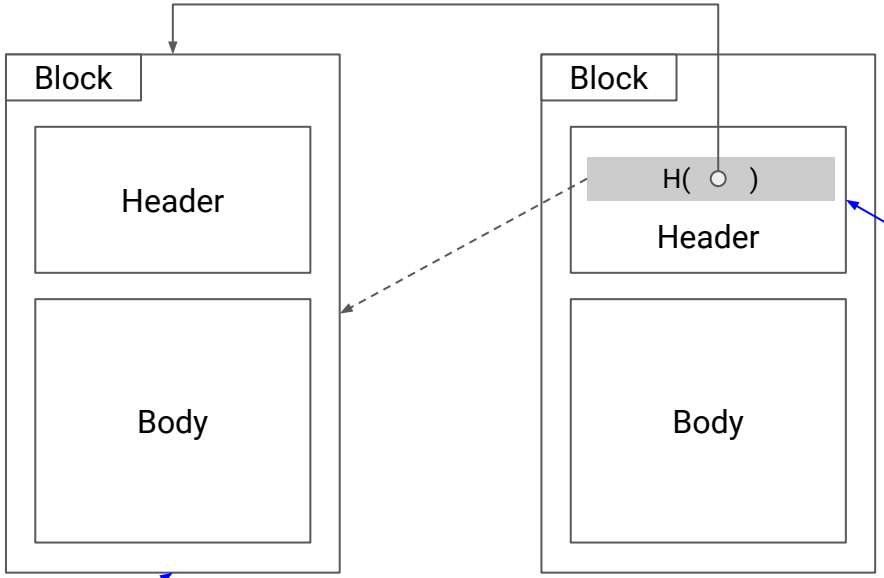
- 문자열 ‘hello!’를 Keccak-256으로 해시한 결과는 다음과 같다:

  `96B8D442F4C09A08D266BF37B18219465CFB341C1B3AB9792A6103A93583FDF7`

같은 데이터라도 다른 함수로 해시할 경우 결과값이 크게 다른 것을 확인할 수 있다.

*두 해시 함수 모두 256 비트 길이의 해시를 생성한다.

블록, 블록헤더, 해시포인터



헤더는 블록을 설명하는 정보와 **이전 블록의 해시**를 포함

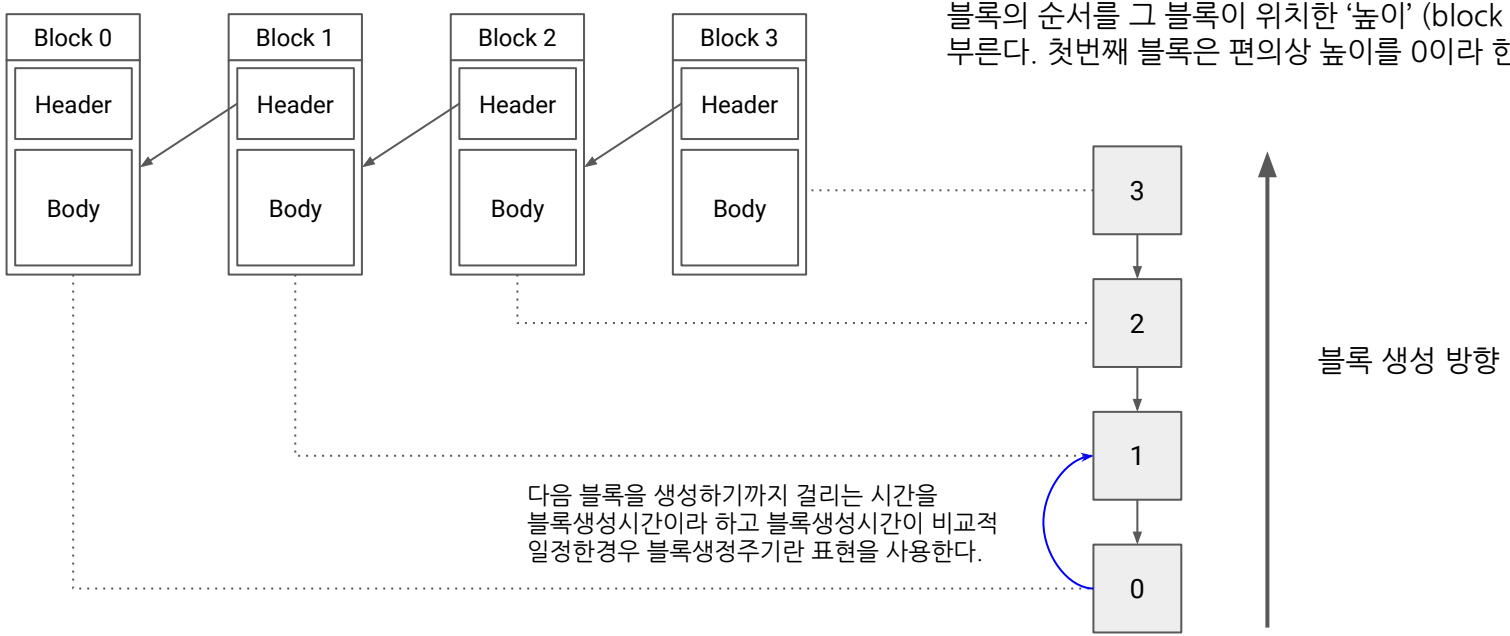
이전 블록의 해시(hash pointer)를 가지기 때문에
(1) 어떤 블록이 앞에 와야는지 결정적으로 알 수 있고
(2) 이를 바탕으로 블록의 순서를 결정할 수 있음

블록은 헤더와 바디로 구분

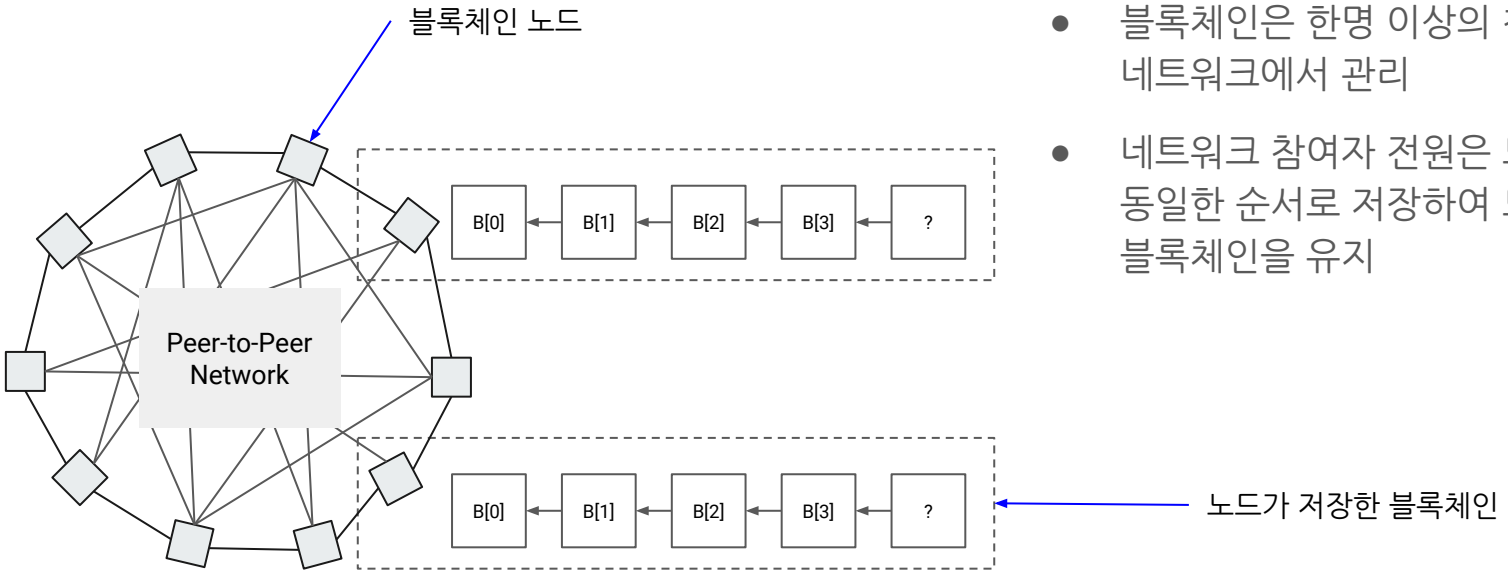
블록높이, 블록생성주기

블록들을 이전 블록이 아래에 최근 블록이 위로 오도록 정렬하면 블록이 생성됨에 따라 체인의 높이가 늘어난다.

블록의 순서를 그 블록이 위치한 '높이' (block height)라 부른다. 첫번째 블록은 편의상 높이를 0이라 한다.

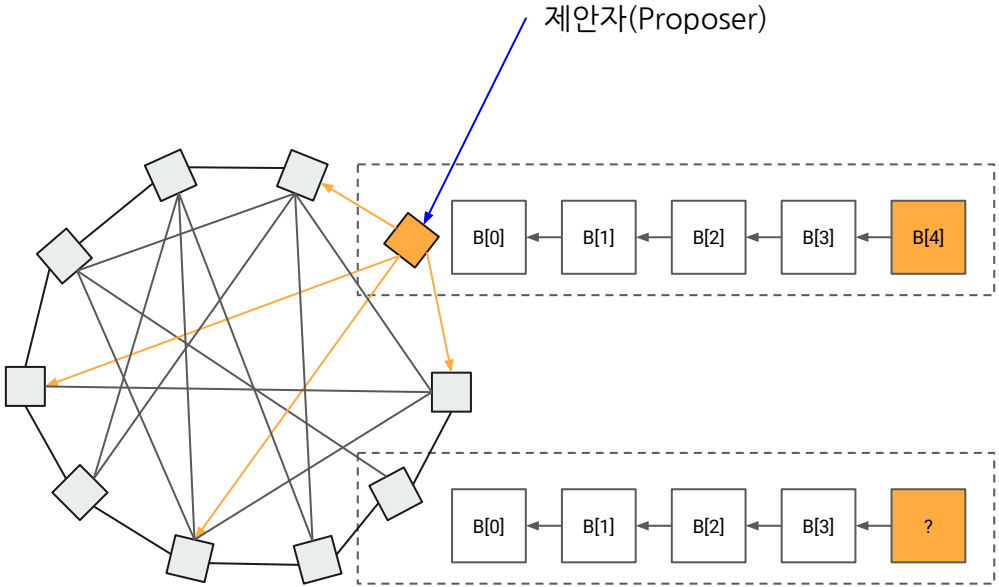


블록체인 네트워크, 노드



- 블록체인은 한명 이상의 참여자가 있는 네트워크에서 관리
- 네트워크 참여자 전원은 모든 블록을 동일한 순서로 저장하여 모두 같은 블록체인을 유지

합의 (Consensus)



- 자격이 있는 참여자는 블록을 제안(propose)할 수 있음
- 블록 제안 자격은 네트워크마다 상이 (e.g., PoW → puzzle solving)
- 블록이 체인에 추가됨 = 참여자들이 새 블록을 자신의 체인에 추가
- 노드들은 제안자가 올바른 자격을 취득했는지, 제안된 블록이 올바른지 **검증** 뒤 블록을 자신의 체인에 추가
- 정족수 또는 정해진 기준을 만족하는 수의 노드가 블록을 자신의 체인에 추가하면 합의가 이뤄졌다고 판단

정리: 블록체인의 불변성과 투명성

- 블록체인은 한명 이상의 참여자가 있는 네트워크에서 관리
- 네트워크 참여자 전원은 모든 블록을 동일한 순서로 저장하여 모두 같은 블록체인을 유지
- 자격이 있는 참여자는 블록을 제안할 수 있음; 블록 제안 자격은 네트워크마다 상이
- 블록이 체인에 추가됨 = 참여자들이 새 블록을 자신의 체인에 추가
- 따라서 새로운 블록이 체인에 추가되려면 네트워크의 합의가 필요; 합의방법은 네트워크마다 상이
    - 어느 한 주체가 단독으로 결정하는 구조가 아닌, 여러 참여자가 합의를 통해 결정하기 때문에 블록체인은 탈중앙화되어 있다고 표현
- 참여자 전원은 이전 블록들을 저장하고 있으므로 새로운 블록의 무결성을 확인가능
- 새롭게 제안되는 블록은 참여자들이 검증 및 합의할 수 있는 형태여야 함 (투명성)
- 한번 쓰여진 블록은 이전의 합의를 번복할 수 있지 않는 한 변경될 수 없음 (불변성)

합의 알고리즘 비교분석

	PoW	PoS	BFT-variants
제안자격 취득 방법	계산이 어려운 문제를 풀 것	플랫폼 토큰을 보유한 양과 기간에 따라 결정적으로 또는 확률적으로 뽑힐 것	정해진 순번 또는 정해진 확률에 의해 뽑힐 것
네트워크 참여 제한	없음	없거나 낮음	높음
합의에 필요한 연산량	높음	낮음	낮음
위협	전체 연산량의 51%를 한 참여자가 소유할 경우 중앙화 됨	전체 토큰의 51%를 한 참여자가 소유할 경우 중앙화 됨	전체 참여노드의 ⅓ 이상이 담합할 경우 합의 불가, 전체 참여노드의 ⅔ 이상이 담합할 경우 중앙화 됨
대표적인 블록체인	Bitcoin, Litecoin, Ethereum, Monero, QTUM	Ethereum FFG & CFG, EOS (dPoS)	Klaytn, Tendermint, Hyperledger Fabric, Ontology

Public vs. Private

Disclaimer: this is a subjective matter

퍼블릭과 프라이빗의 구분은 블록체인에 다음을 수행할 수 있는지 확인하여 결정:

- 누구든지 기록된 정보(블록)를 자유롭게 읽을 수 있는지?
- 명시적인 등록 또는 자격취득 없이 정보를 블록체인 네트워크에 기록할 수 있는지?

블록체인의 정보가 공개되어 있고 네트워크가 정한 기준(e.g., gas fee)에 따라 정보를 기록요청할 수 있다면 그 블록체인은 퍼블릭/공개형이라 한다.

이와 반대로 정보가 공개되어 있지 않고 미리 자격을 득한 사용자만이 정보를 기록할 수 있다면 그 블록체인은 프라이빗/비공개형이라 한다.

Permissionless vs. Permissioned

Disclaimer: this is a subjective matter

일반적으로 **네트워크의 참여**가 제한된 경우 ‘permissioned’, 그렇지 않은 경우 ‘permissionless’라 정의

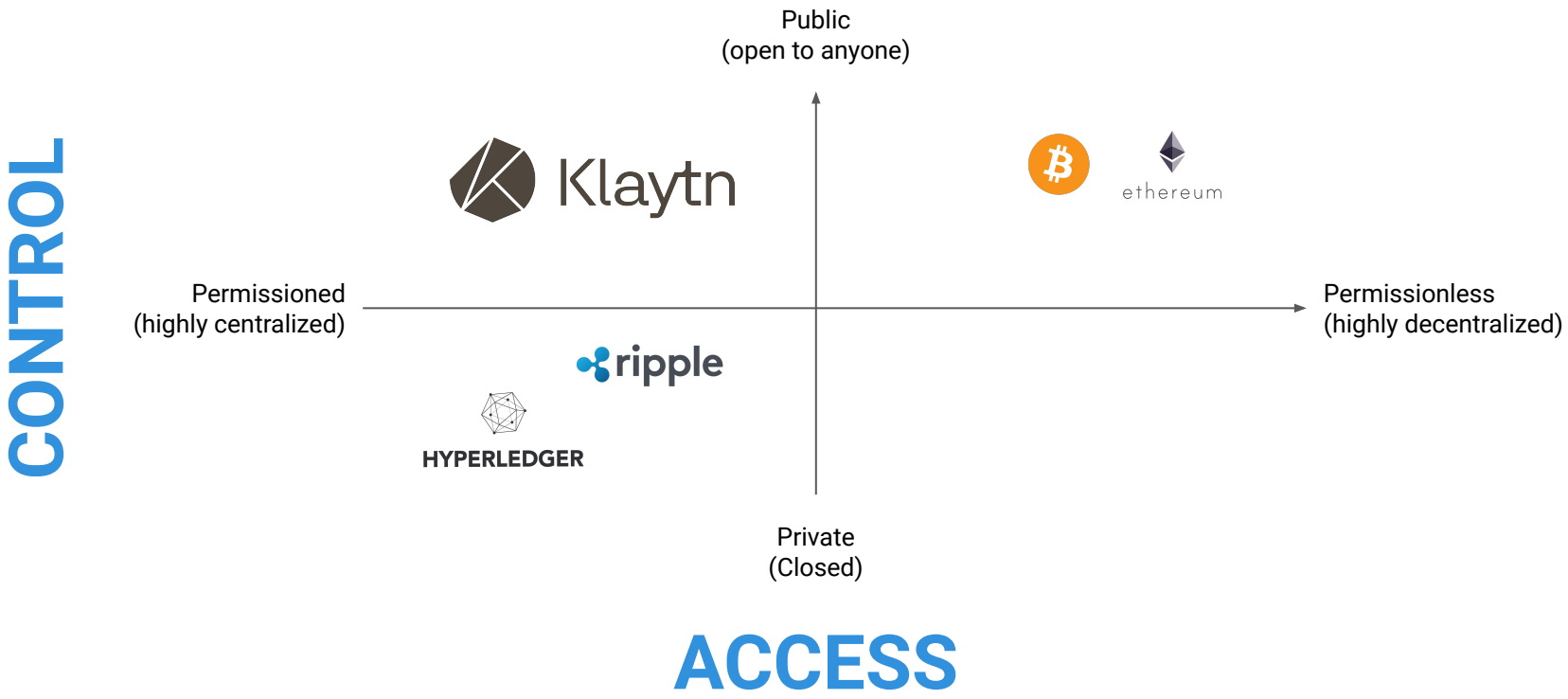
네트워크의 참여의 정의

- (넓은 의미) 블록체인 P2P 네트워크에 참여
- (좁은 의미) 합의과정의 참여

Public/Private의 개념이 **정보의 접근성(Access)**와 관련이 있다면
Permisionless/Permissioned는 **정보의 제어 (Control)**, 즉 무엇이 블록에 포함되는지를 결정하는 지에 더 연관

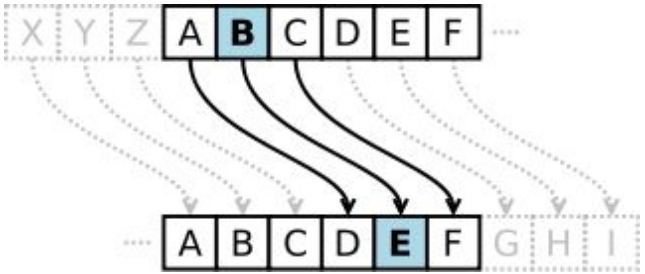
예: Ethereum → Public, Permissionless
Klaytn → Public, Permissioned

유형별 블록체인 비교분석



암호화

고전적인 암호: 카이사르 암호



암호화하고자 하는 내용을 알파벳별로 일정한 거리(distance, d)만큼 밀어서 다른 알파벳으로 치환하는 기법.

암호를 풀거나 만들때 알아야하는 정보를 키(Key)라고 함. 위 예제에서 사용된 키는 d=3.

대칭키암호/비대칭키암호

평문(Plain Text)이란 암호화 되어 있지 않은 문자열을 의미

- **암호화**는 평문을 암호로 만드는 것 (cipher, encrypt)
- **복호화**는 암호를 평문으로 만드는 것 (dechipher, decrypt)

암호화에 사용한 키와 복호화에 사용한 키가 동일한 경우 **대칭키암호**로 분류

암호화에 사용한 키와 복호화에 사용한 키가 다를 경우 **비대칭키암호**로 분류

비대칭키암호 (공개키암호)

두개의 키를 사용하여 암호화와 복호화를 실행

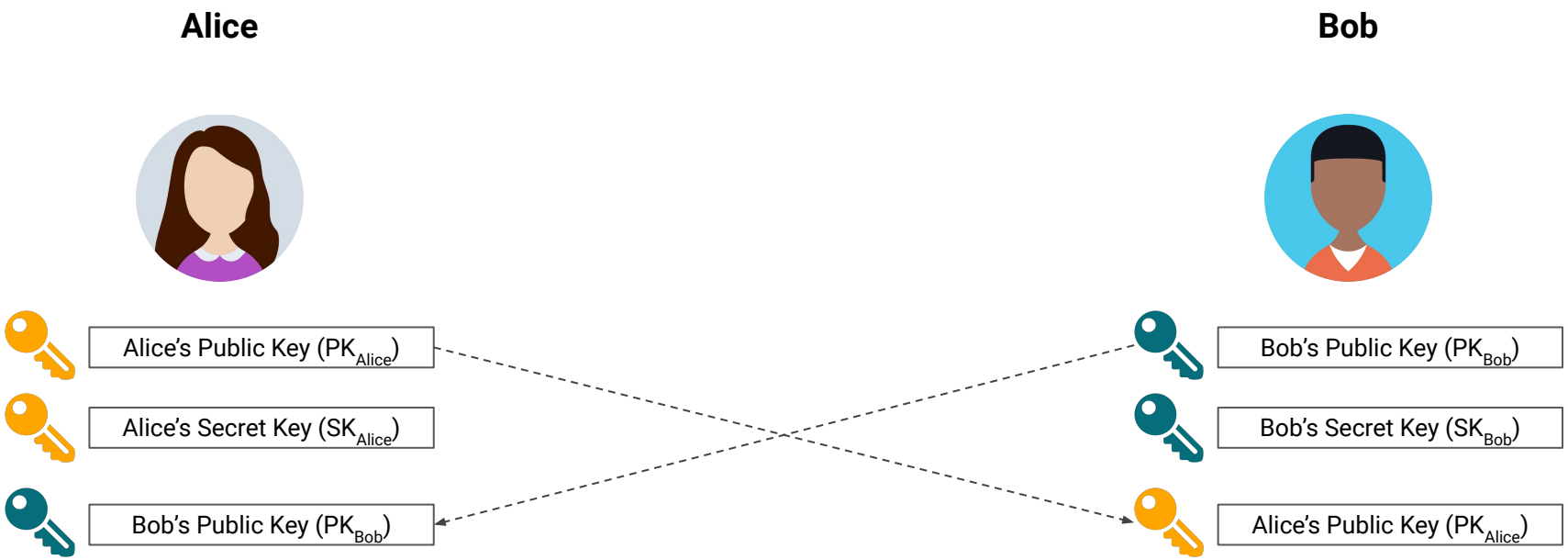
- 암호화에 사용되는 키 = 공개키 (Public Key, PK)
- 복호화에 사용되는 키 = 비밀키 (Private Key/Secret Key, SK)

비대칭키 암호의 목적:

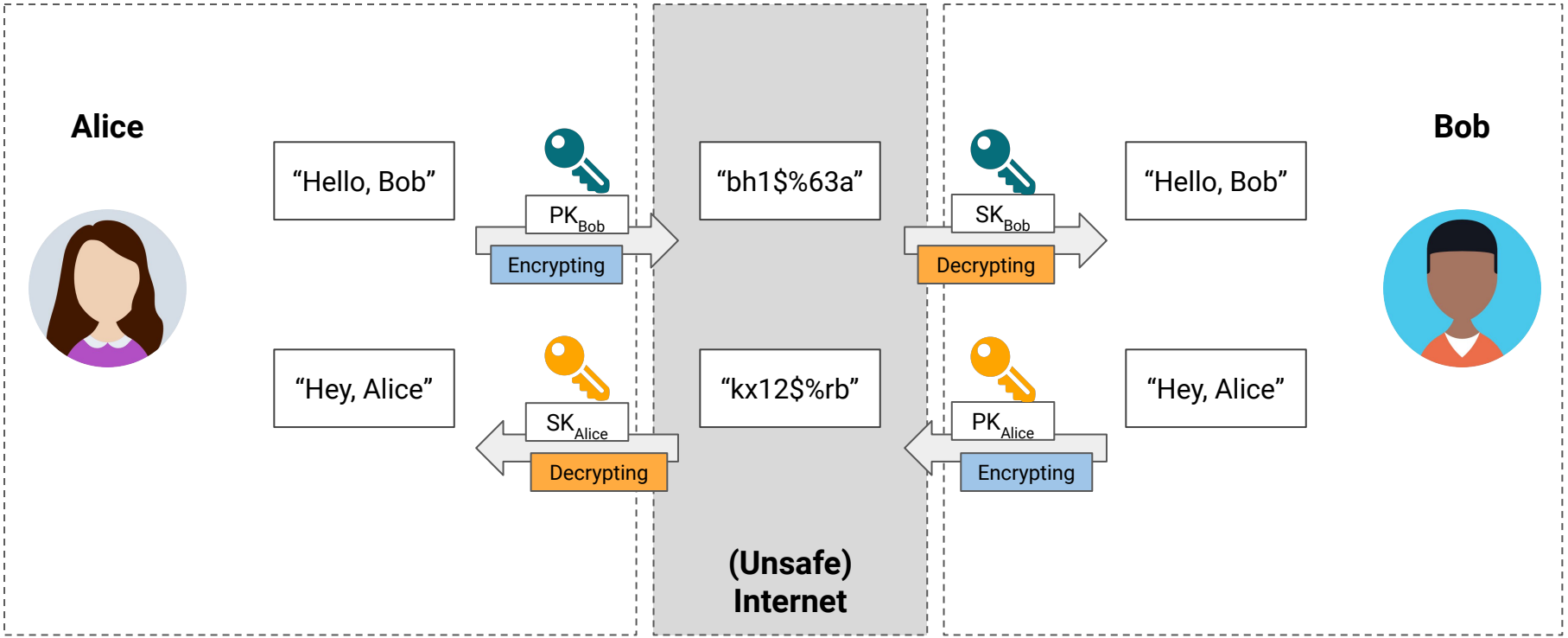
“누구든지 암호화할 수 있지만 **비밀키를 아는 사람**만 복호화할 수 있어야 한다”

- 공개키와 비밀키는 한쌍으로 묶여있는 아주 큰 숫자들
- 비밀키로부터 공개키를 도출하는 것은 쉬움
- 공개키로부터 비밀키를 찾는 것은 매우 어려움

공개키암호를 사용한 안전한 통신



공개키암호를 사용한 안전한 통신



전자서명

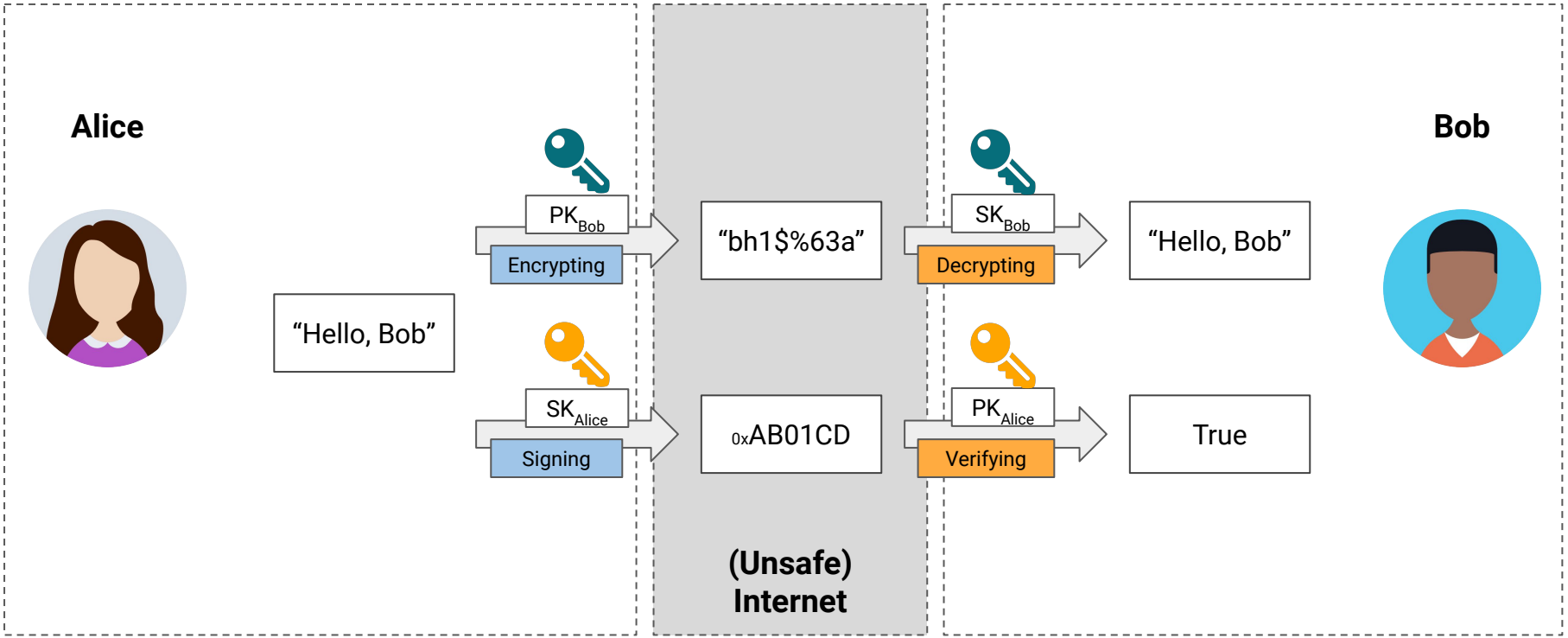
비대칭키암호는 지정된 사람만 정보를 확인할 수 있도록 도움 (privacy)

- Alice가 Bob에게 메세지를 보낼 때 PK_{Bob}을 사용
- Bob은 이 메세지가 Alice에게서 온 것인지 어떻게 확인할까?

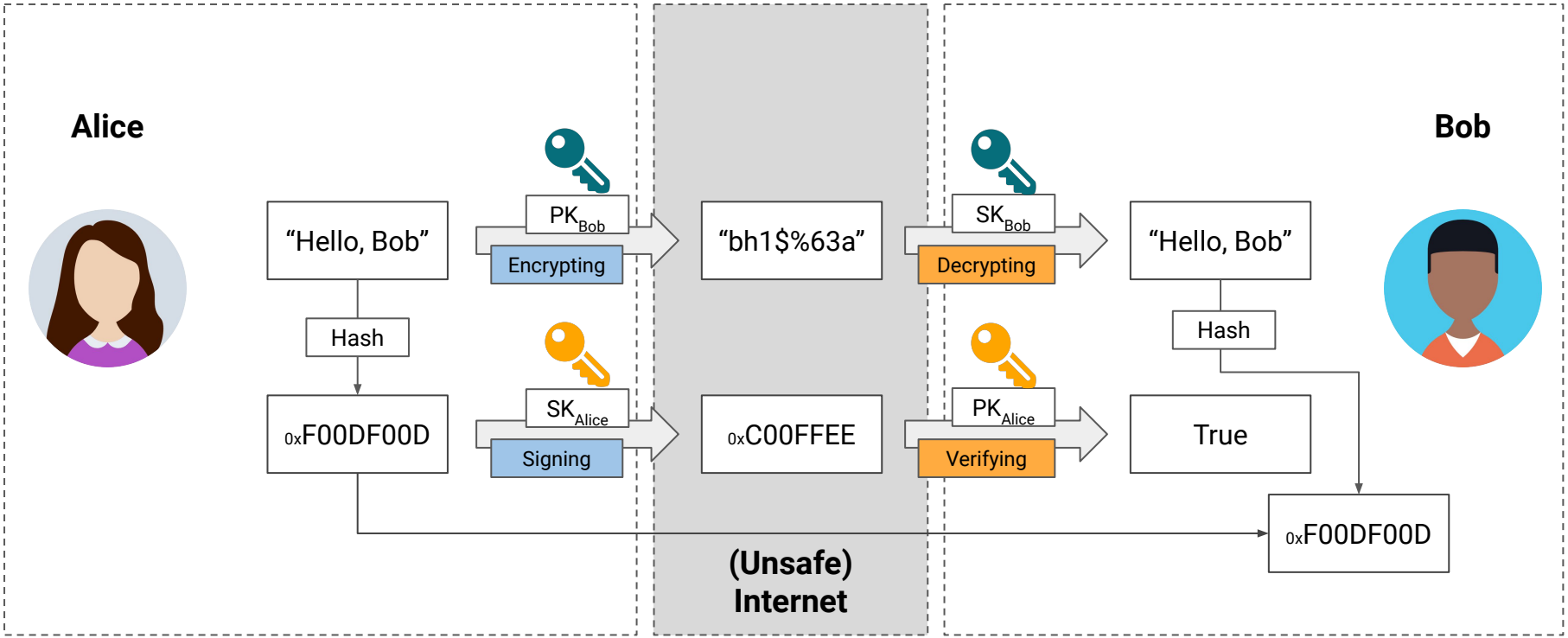
전자서명은 누가 정보를 보냈는지 알기 위해 사용 (non-repudiation)

- 전자서명은 비대칭암호의 응용 프로그램
- 서명은 비밀키로만 생성가능
- 공개키는 서명이 짝을 이루는 비밀키로 생성되었는지를 검증

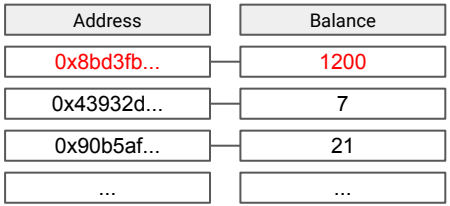
공개키암호와 전자서명을 사용한 안전한 통신

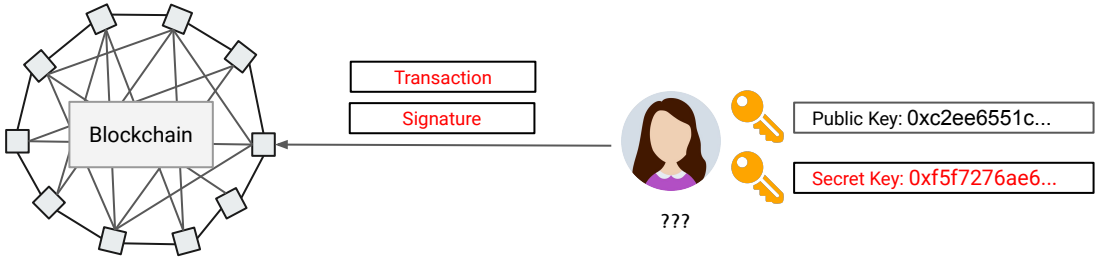


공개키암호와 전자서명을 사용한 안전한 통신



블록체인과 공개키암호

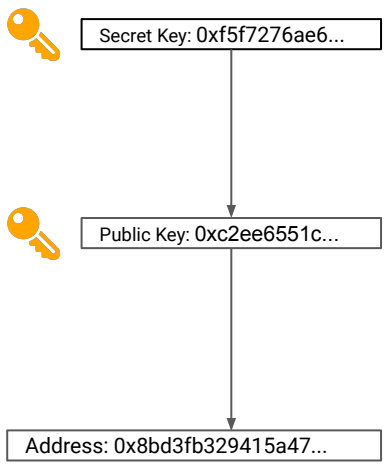




- 블록체인은 암호학적 기법을 토대로 만들어진 기술
- Bitcoin은 네트워크 참여자 모두가 같은 '원장'을 공유함으로써 투명한 거래가 가능
- 원장은 어느 주소에 BTC가 있는지 기록하지만 그 주소가 누구에게 속하는지는 기록하지 않음 (anonymity)
- Bitcoin은 공개키암호를 사용하여 **명시적인 비밀교환과정 없이** BTC의 소유권 증명을 실행
- 누구든 해당 주소로 변환가능한 공개키로 검증가능한 서명을 생성할 수 있다면 그 주소의 제어권을 소유하고 있다고 가정

공개키암호화를 사용한 소유권 증명

- 대부분의 블록체인 주소는 공개키로부터 도출된 값
  - Bitcoin: Hash160 of a public key where Hash160 = RIPEMD160 + SHA256
  - Ethereum: Rightmost 160 bits of Keccak hash of a public key
- Bitcoin의 경우
  - 임의의 주소 X에 10 BTC가 있다고 가정할 때 Alice는 X에서 또다른 임의의 주소 Y로 5 BTC를 전송(i.e., transfer 5 BTC from X to Y)하는 거래를 성사시키기 위해 X로 변환되는 공개키와 짝을 이루는 비밀키로 해당 거래를 서명할 수 있어야 한다.
- Ethereum의 경우
  - 임의의 주소 X에 위치한 어카운트의 잔고에 10 ETH가 있다고 가정할 때 Alice는 X에서 또다른 임의의 주소 Y에 위치한 어카운트로 5 ETH를 전송(i.e., transfer 5 ETH from the account at X to the account at Y)하는 거래를 성사시키기 위해 X로 변환되는 공개키와 짝을 이루는 비밀키로 해당 거래를 서명할 수 있어야 한다.



구현 방법으로 나눠본 블록체인

UTXO (Unspent Transaction Output) 기반 블록체인

- 블록체인에 사용 가능한 토큰(e.g., Bitcoin) - UTXO들과 사용 자격검증방법을 기록
- 일반적인 자격검증방법은 UTXO의 정보와 일치하는 공개키로 검증가능한 전자서명을 제출하는 것
- Bitcoin이 대표적인 UTXO 기반 블록체인

어카운트 기반 블록체인 (Account-based Blockchain)

- 어카운트는 블록체인을 구성하는 주체(entity)를 표현하며 상태를 기록
- 사용자는 어카운트를 사용할 때마다 어카운트 공개키로 검증가능한 전자서명을 생성
- 상태를 기록할 수 있기 때문에 스마트 컨트랙트를 구현하기에 용이
- Ethereum, Klaytn이 대표적인 어카운트 기반 블록체인

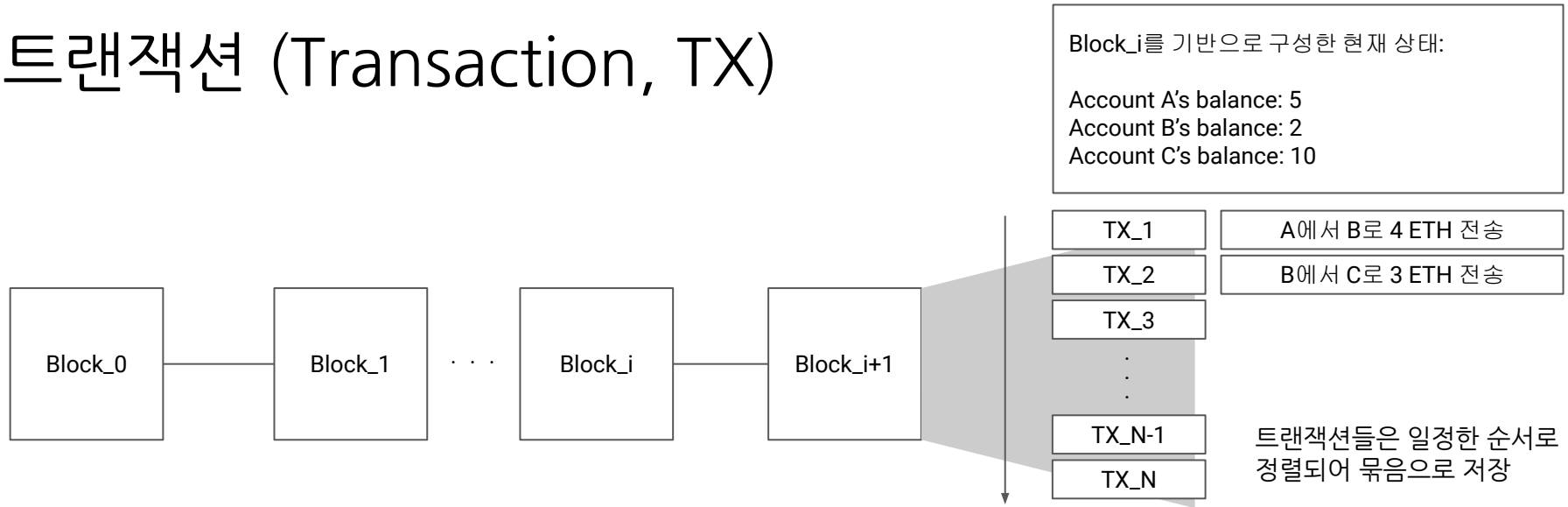
Ethereum 어카운트, 주소, 상태

- Ethereum의 어카운트는 Ethereum의 주체(entity)를 표현하고 그 상태를 기록하는데 사용
- 어카운트는 **EOA(Externally Owned Account)**와 **스마트 컨트랙트**로 구분
- Ethereum 사용자는 EOA를 사용
- 사용자는 임의의 공개키와 비밀키 쌍(Key Pair)을 생성한뒤 공개키를 어카운트 주소로 변환하여 EOA를 생성
    - 별도의 승인과정이 필요없으며 Ethereum 네트워크와 통신도 필요없음
    - 위 과정으로 인해 어카운트는 특정 키페어에 종속
- 사용자의 상태(state)는 어카운트 주소로 찾을 수 있는 블록체인 저장공간에 기록

비밀키	공개키	주소
0x5db23f5fa5c7e57808f14a50473d6f32808065052c25457bde4e413c3508cc39	0x01e89b34ea5ca3fc6e33eec3011629163190700d02a3e5b4dbf248de2386da7a289e174b547883fd6c5a54608545caf1dc896dc0f214b4d0ab7f45324b095857	0x304e70ee5ee2c2ed23ca3ea9b07034a67f0b56d6

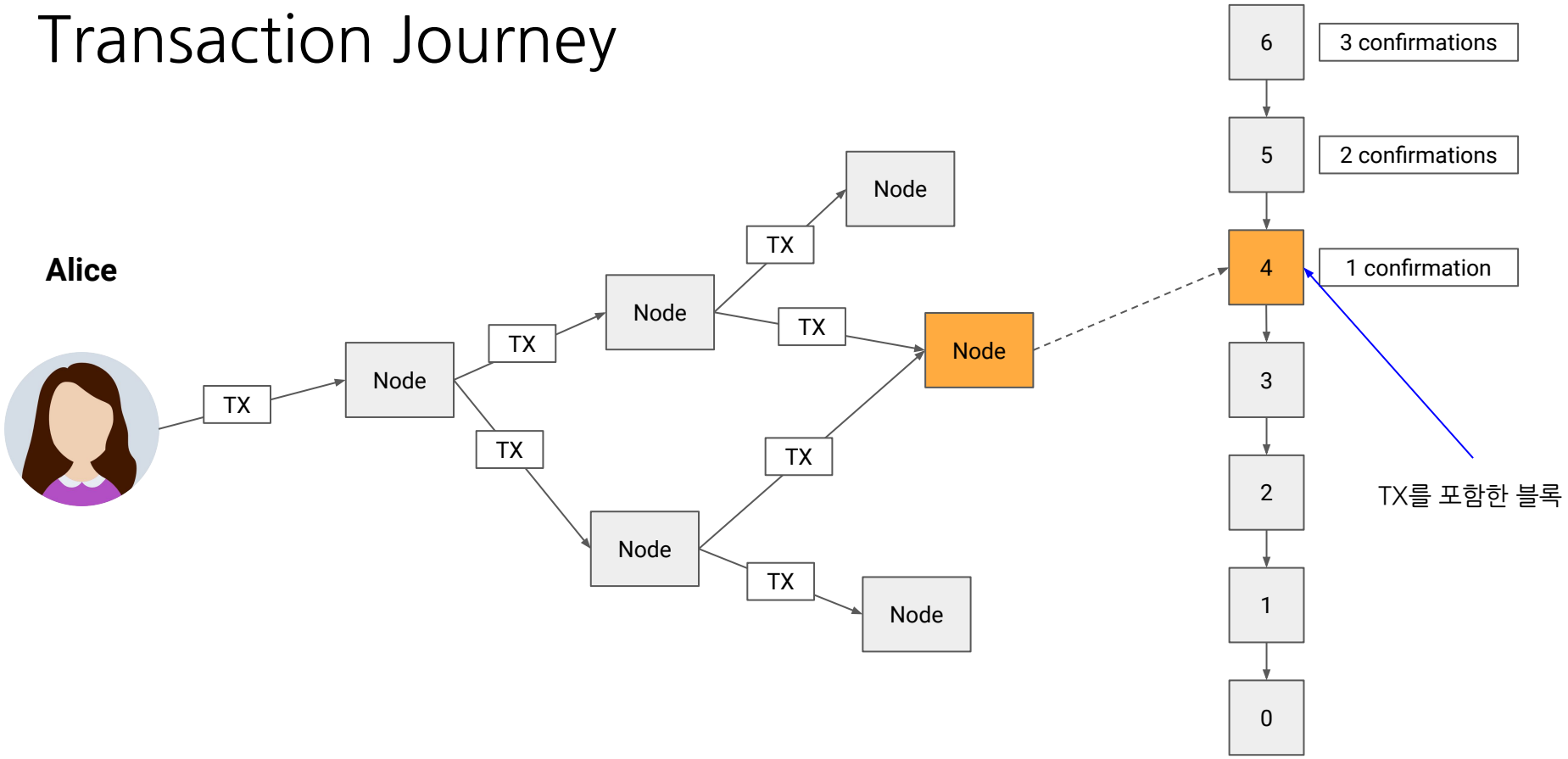
https://blockchains.tools에서 확인 가능

트랜잭션 (Transaction, TX)



- 블록은 트랜잭션들을 일정한 순서로 정렬하여 저장하는 컨테이너
- 트랜잭션은 어카운트의 행동
- 트랜잭션의 순서는 중요; TX_1 → TX_2으로 진행하는 것은 괜찮지만 TX_2 → TX_1로 진행하는 것은 불가능
- 블록체인 참여자들은 블록을 검증할 때 트랜잭션들이 올바른 순서대로 정렬되었는지를 확인 후 합의
- 각각의 트랜잭션들은 어카운트에 연결된 공개키로 검증가능한 서명을 포함

Transaction Journey



Confirmation vs. Finality

- Confirmation 숫자는 트랜잭션이 블록에 포함된 이후 생성된 블록의 숫자
  - 임의의 트랜잭션 T가 포함된 블록의 높이가 100, 현재 블록높이가 105라면 T의 confirmation 숫자는 6
- PoW를 사용하는 블록체인들은 finality가 없기 때문에 confirmation 숫자가 중요
- Finality란 블록의 완결성을 의미
  - 합의를 통해 생성된 블록이 번복되지 않을 경우 완결성이 존재
- PoW 기반 합의는 확률에 기반하기 때문에 경우에 따라 블록이 사라질 수 있으므로 완결성이 부재함
  - PoW 블록체인은 수학적으로 복잡한 퍼즐을 풀어 블록을 제안할 자격을 얻는 구조
  - 만약 두명의 서로 다른 참여자가 동시에 퍼즐을 풀어 두개의 올바른 블록을 생성한다면 두 블록 중 하나는 (eventually) 사라지게 됨
  - 이 때문에 블록이 확률적 완결성를 갖기까지 일정 갯수 이상의 블록이 생성되기를 기다려야 함
  - Bitcoin, Ethereum 모두 longest chain (or heaviest chain) 법칙을 사용

Understanding Bitcoin's 6 Confirmations Rule

- 네트워크 시차로 인해 생성된 우연한 복수의 블록들 가운데 하나가 선택되는데 필요한 블록은 두어개 정도 → 2~3 confirmations
- 퍼즐을 빠르게 풀 수 있는 악의적인 참여자(공격자)가 있을 경우 그 참여자의 해시능력(hash power)에 따라 필요한 confirmation 숫자가 달라짐
    - 해시능력이 높을 수록 퍼즐을 푸는 속도도 빠르기 때문에 주어진 문제를 먼저 풀 확률이 높아짐
    - 해시능력이 높은 참여자는 longest chain을 임의로 선택 또는 생성할 수 있음
    - 따라서 해시능력을 감안하더라도 임의로 블록체인을 변경하지 못할 정도로 충분히 많은 블록이 생성되기를 기다려야할 필요가 생김
    - Bitcoin의 6 confirmation 법칙은 공격자가 전체 해시능력의 약 25%를 가질 때를 가정한 숫자

Confirmations to wait if...

PoW 블록체인이 공격자가 전체 해시능력의 x%를 가질 때 99.9% 완결성을 확보하려면 다음과 같은 confirmation 숫자가 필요:

x	# confirmations	기다려야하는 시간
10	4	40분
25	7	1시간 10분
33	10	1시간 40분
49	170	2시간 50분

BFT 기반 블록체인

- BFT 기반 블록체인은 블록의 완결성이 보장됨
  - 네트워크가 동기화되어 있기 때문
  - 블록 생성이 PoW에 비해 빠르고 경제적
- 하지만 네트워크 동기화의 필요로 인해 참여자의 숫자가 제한됨
  - 네트워크 참여자 구성이 고정되어 있어야 합의가 가능
  - 구성이 변경될 경우 모든 네트워크 참여자가 새로운 구성을 인지하기 까지 합의 불가능
  - 합의 알고리즘이 네트워크 동기화를 가정하고 짜여졌기 때문에 네트워크 사용량이 높음
  - 참여자가 많아질 경우 네트워크 오버헤드로 인해 합의가 느림

End of Document