비트코인 개념과 핵심기술

KAIST 디지털금융전문가 과정

IFC 여의도

이흥노, GIST, South Korea
Home page: http://infonet.gist.ac.kr
Github: https://github.com/cryptoecc
Facebook/Publication ID: Heung-No Lee
E-mail: heungno@gist.ac.kr

디지털금융전문가과정

DIGITAL FINANCE MASTERSHIP PROGRAM

2021 봄학기

- **블록체인과 디지털 자산**
  금융혁신과 신사업준비를 위하여 블록체인 분야에 특화된 기술 교육이 필요한 자
- **클라우드 컴퓨팅과 빅데이터 분석**
  금융 분야에서 클라우드 컴퓨팅 기반 기술 적용과 빅데이터 기획, 분석, 활용을 통한 가치 창출에 관심있는 금융 관련 종사자

2021 가을학기

- **디지털 트랜스포메이션**
  디지털금융/핀테크 분야 백그라운드가 없는 전통 금융 종사자 대상
- **인공지능과 기계학습**
  인공지능과 기계학습 분야를 중점적으로 이수하여 금융 산업에서 해당 분야의 전문가로 성장 하고 싶은 자

KAIST 정영대학

디지털금융전문가과정 Digital Finance Mastership Program

07326 서울특별시 영등포구 국제금융로 10 KAIST 디지털금융전문가과정
(One IFC 17층 KAIST 이의도 캠퍼스)
T. 02) 6274-6181~2 F. 02)6274-6194 E.kaistdfmp@kaist.ac.kr
www.business.kaist.ac.kr/executive

협력기관

서울특별시 금융위원회

QR 바로가기

440 x 280 mm

디지털금융전문가과정

BLOCKCHAIN & DIGITAL ASSETS

Digital Finance Mastership Program

2021년 교육일정 (2021. 3. 17 ~ 7. 14)

KAIST 정영대학

이흥노 교수 강의

교육 일자	일자	시간	시간	강의 주제	강사진	모듈
03월 31일	금	19:00~ 22:00	3H	블록체인 이코노미	이흥노 (GIST)	디지털금융과 블록체인
04월 14일	수	19:00~ 22:00	3H	비트코인 개념과 핵심 기술	이흥노 (GIST)	블록체인 플랫폼과 서비스

Agenda of this talk

- Bitcoin Consensus
- Other Consensuses
- Other Approaches

보유기술현황

보유기술

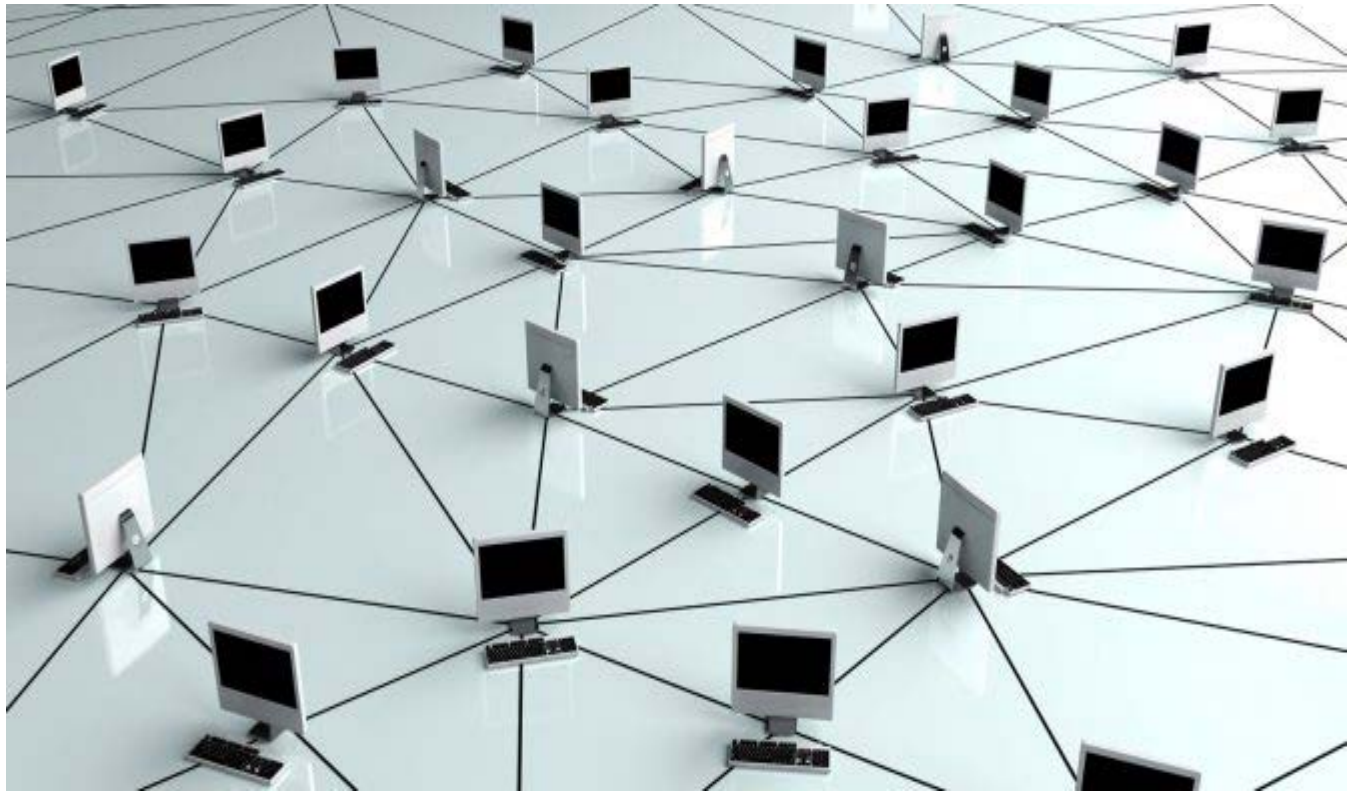
- GIST는 블록체인의 재중앙화 문제 해결을 위한 **부호-암호 기반의 작업증명** 방식 연구 및 특허 확보 및 출원한 **부호-암호 작업증명 특허**
  (GIST IP)를 기반으로 블록체인 시스템 개발

특허

특허	출원인	출원번호/등록일	핵심기술
부호-암호 화폐 시스템	광주과학기술원	10-2019-0151246 / 2019.11.12	블록체인의 재중앙화 문제 해결을 위한 핵심 기술
블록체인거버넌스	광주과학기술원	10-2019-0084800 / 2019-07-12	블록체인 거버넌스와 규제
블록체인의 거래검증시스템, 및 블록체인이 거래검증방법	광주과학기술원	10-2019-0120655 / 2019-09-30	블록체인거래검증

비트코인은 인터넷 위에서 동작

P2P 노드 네트워크 구성



멀리 떨어진 두 사람간의 거래

➢지급결제거래: A가 B에게 2 BTC를 보낸다.

A → B 2 BTC

미국에 있는
Adam

한국에 있는
Byungchul

Consensus

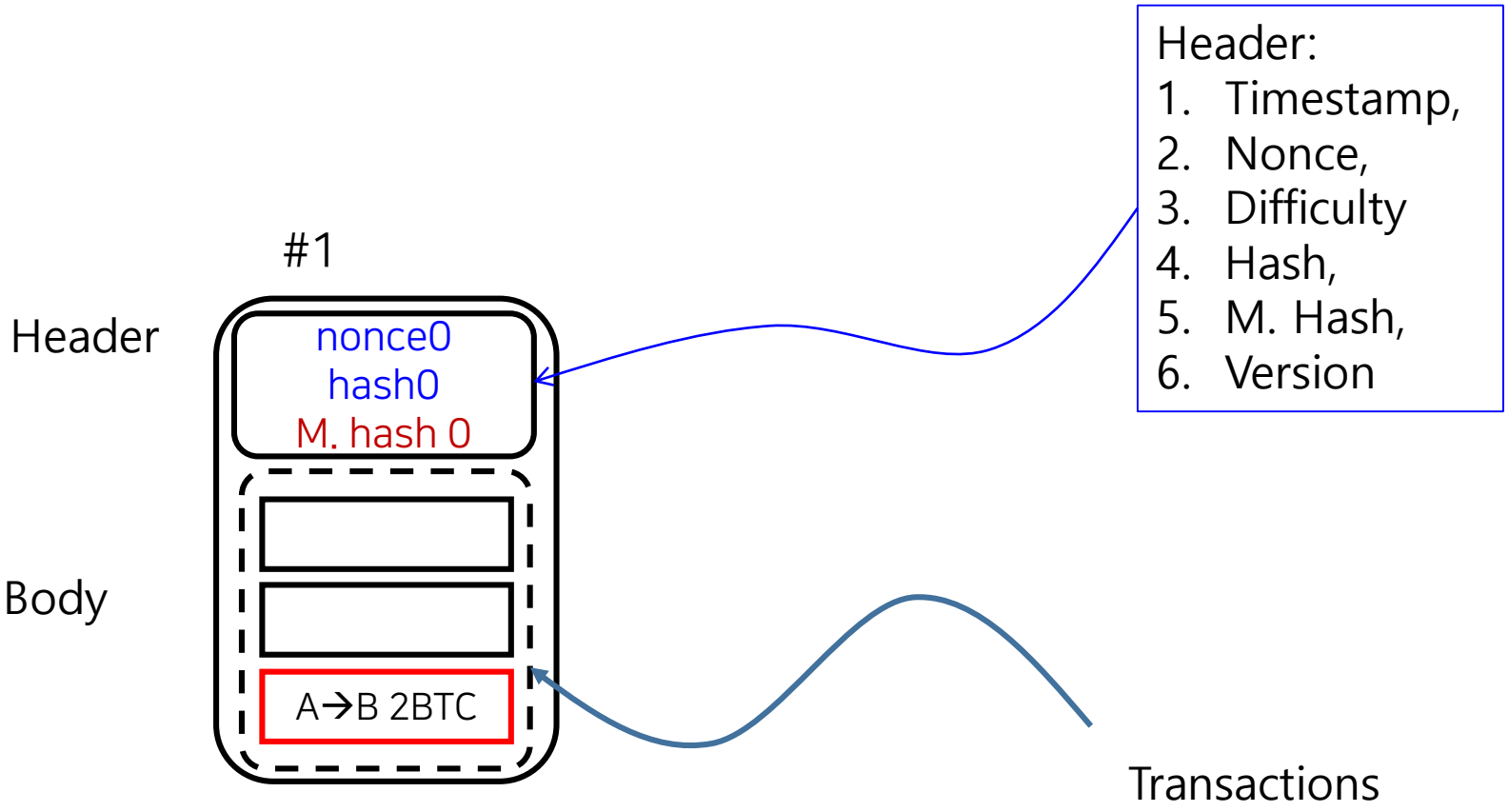
- **Goal: 하나의 동일한 원장 유지 관리**

- Nodes: 글로벌 영역에 분산되어 있는 노드 N개
  1. State: 각 노드의 현재 상태
     A. Work: (각자) 새로운 TXs 담아 블록 생성
     B. Announce: 발표
     C. Inspection: 검사
     D. Approval: 승인
  2. State Update: 새로운 블록을 기존 블록체인에 순서대로 추가

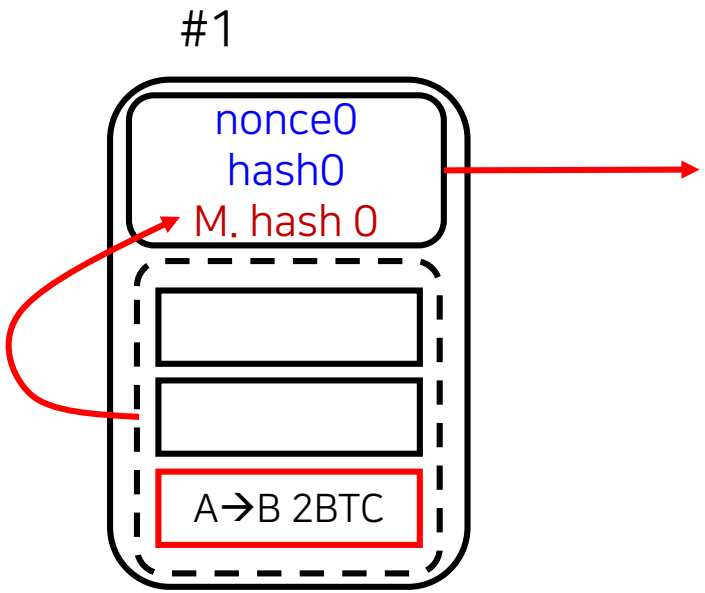
Consensus Algorithm설계의 어려움

- 노드들은 인터넷 상에서 협력하여 합의에 도달해야 함
- Open Source SW
- Cryptocurrency의 경우 다양한 공격 방법이 존재
  - Sybil Attack
  - DDoS Attack
  - Byzantine faults

Block: Header 와 Body



Secure Hash Function: 블록요약을 만드는 함수



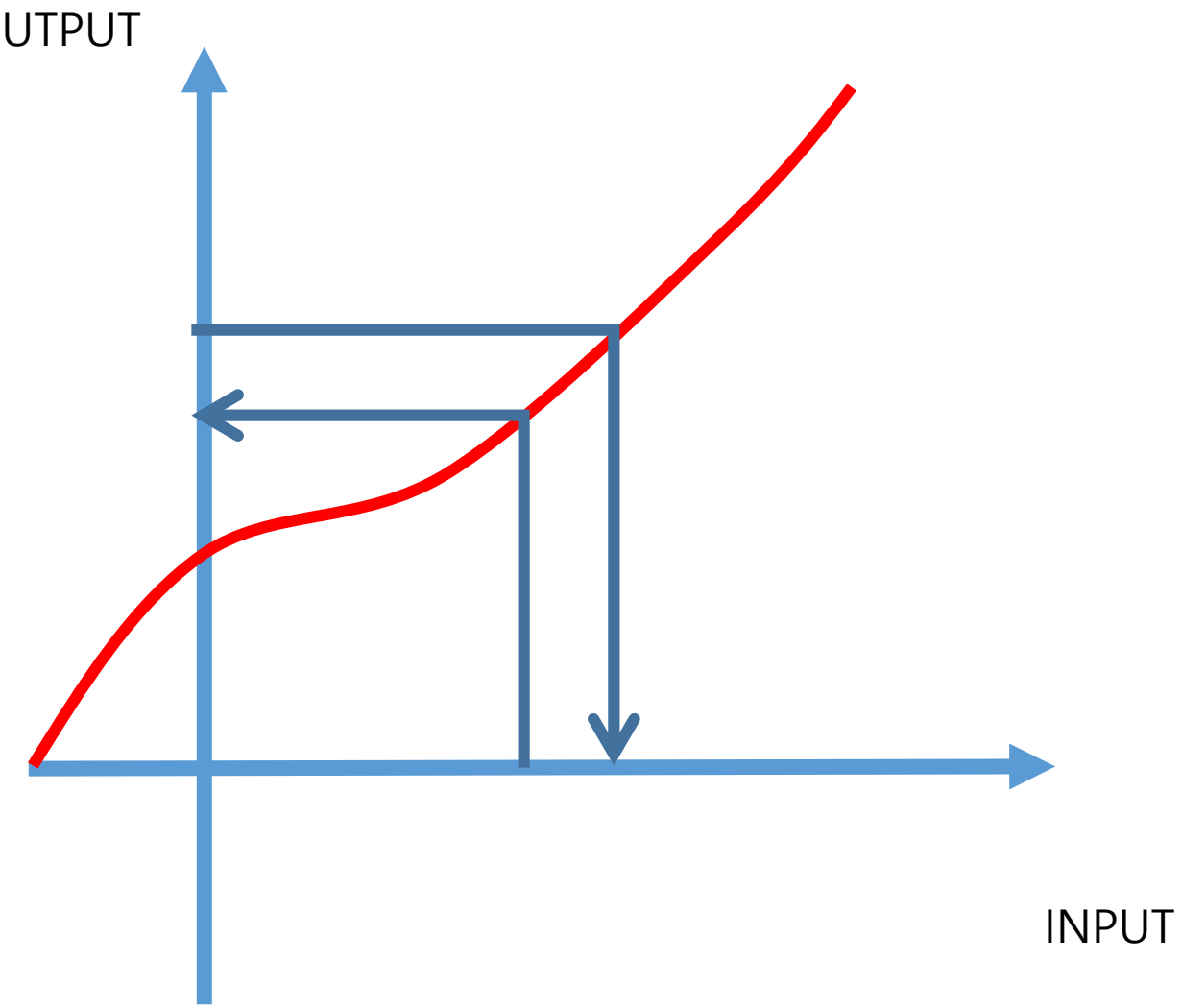
특징: Oneway 앞으로만 가는 함수.
뒤로 가려면 대입법 밖에 없다.

함수는 INPUT 을 받고 OUTPUT을 만든다
INPUT: 파일
OUTPUT: 256 bit 숫자

쓰이는 곳
1. 은행 서버는 Passwd의 hash값만 저장해 놓는다.
2. Spam mail 방지
3. PoW

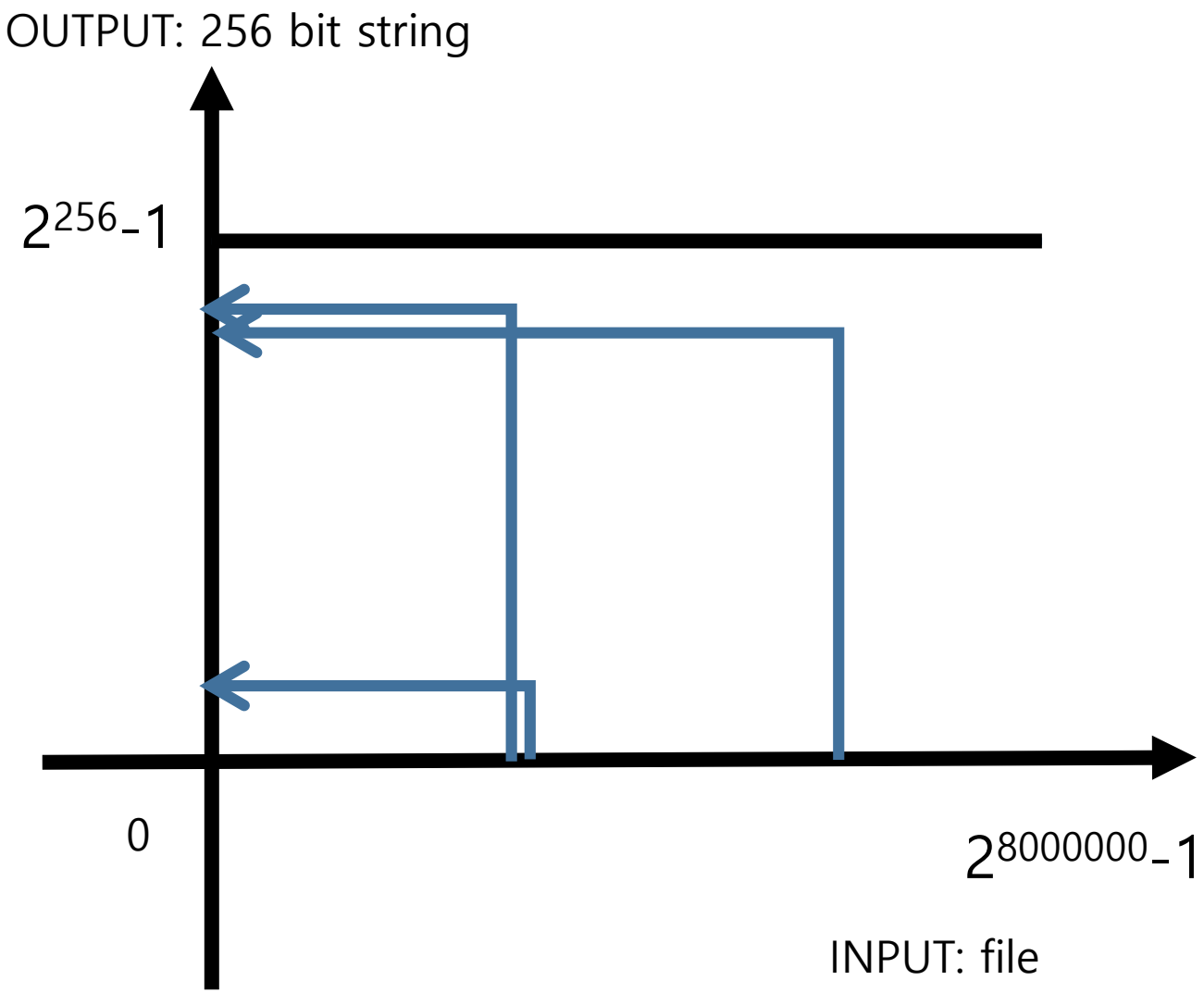
일반 함수

- Output을 본 후,
  Input 예측 가능함



SHA 함수

- 동일한 Input은 언제나 동일한 Output 출력.
- Input이 조금만 차이가 나도, 마치 무작위로 선택된 것 같은 Output을 줌.
- 결과: Output 값을 얻은 후 Input값을 전혀 예측 할 수 없게 됨.
- Many to 1 함수, but collision free!



Modulo Operation

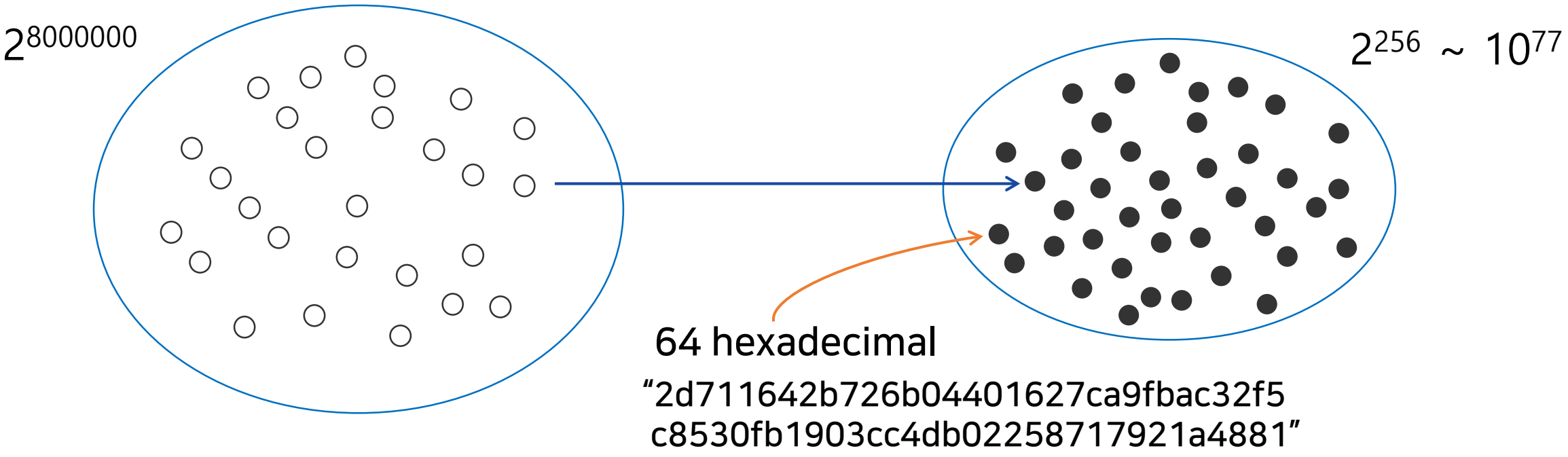
- F(x) = y
- F(x) = 7x + 6 mod 5

- Let x = 5. Then, y = 7*5 + 6 = 41 mod 5 = 1.
- Let x = 4. Then, y = 7*4 + 6 = 34 mod 5 = 4.

Hash 함수

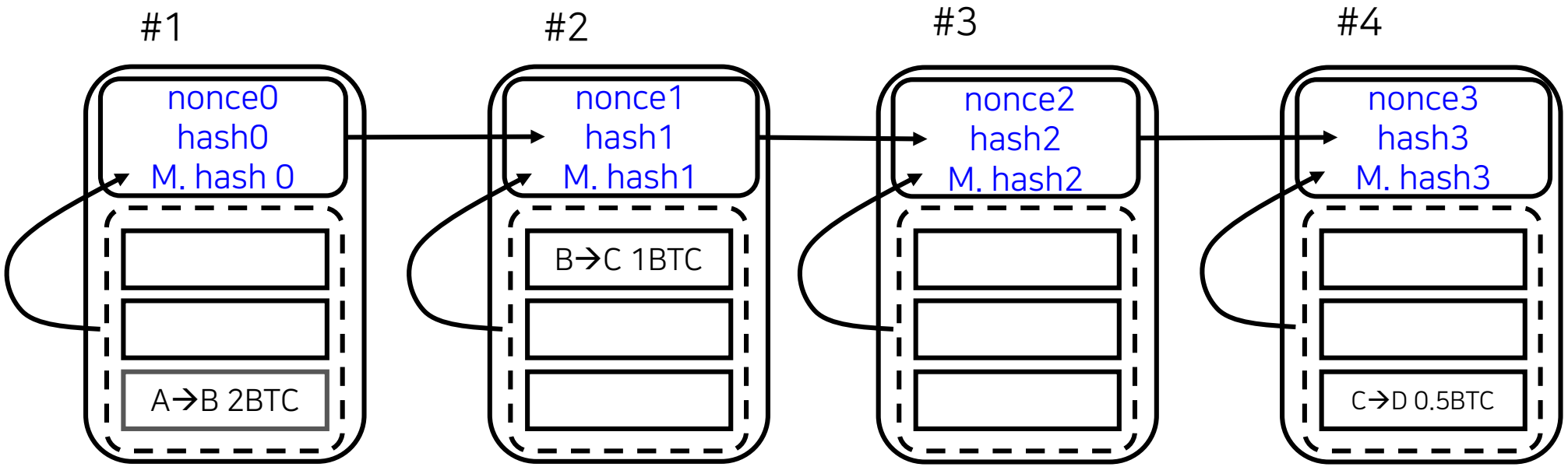
- SHA256, $F(x) = y$

$X = \{x | x$ is a message up to 1 Mbyte in size$\}$ $Y = \{y | y$ is a 256bit string$\}$

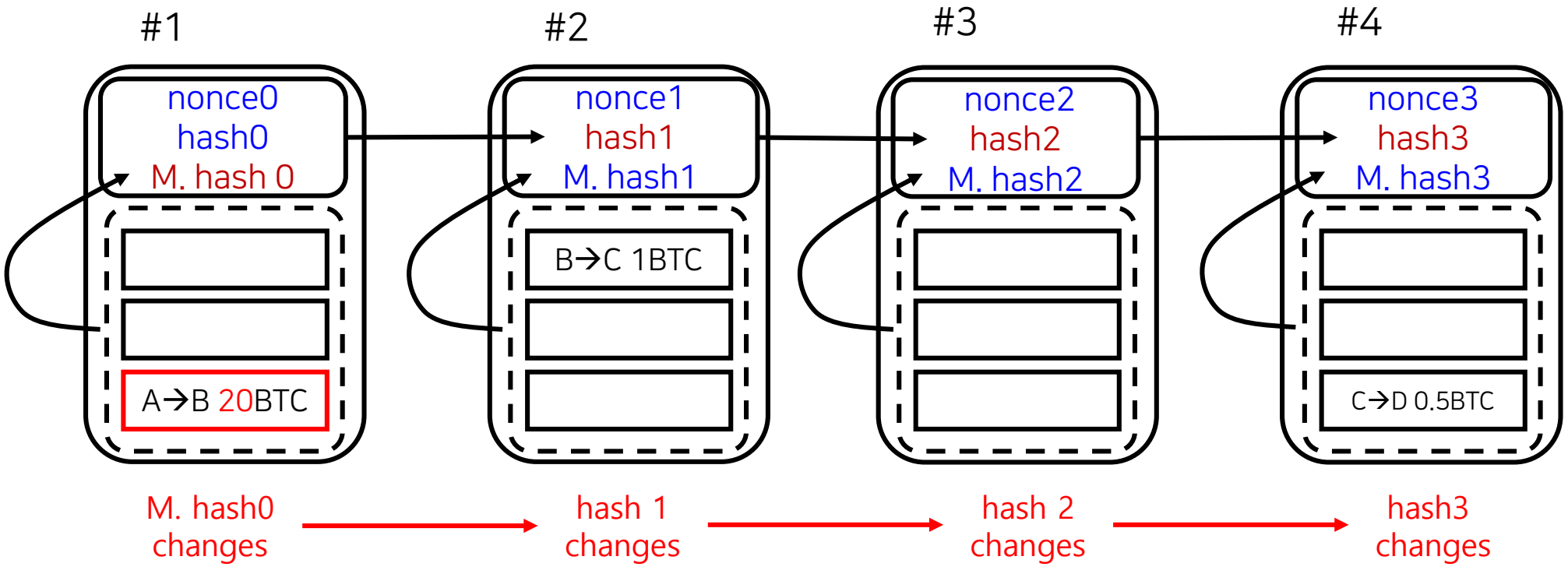


Blockchain이란?

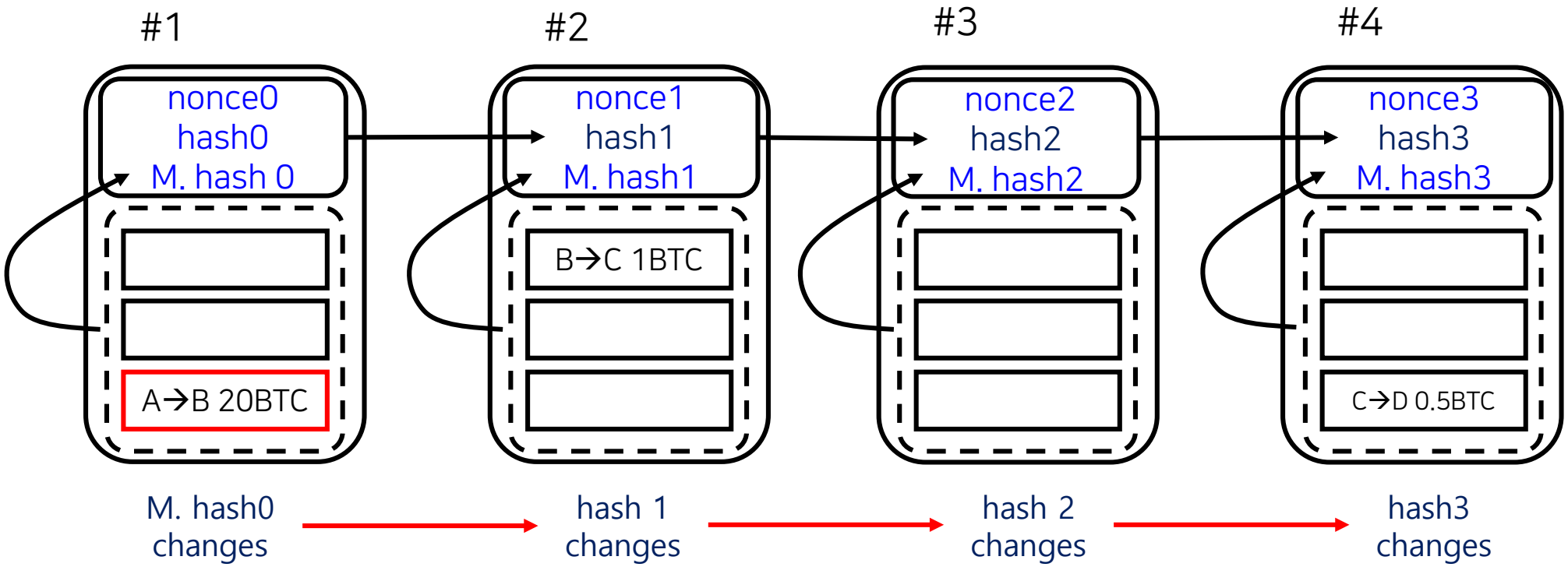
- Block body의 hash값을 Block header에 넣고
- Oneway function으로 block 과 block을 연결하는 것



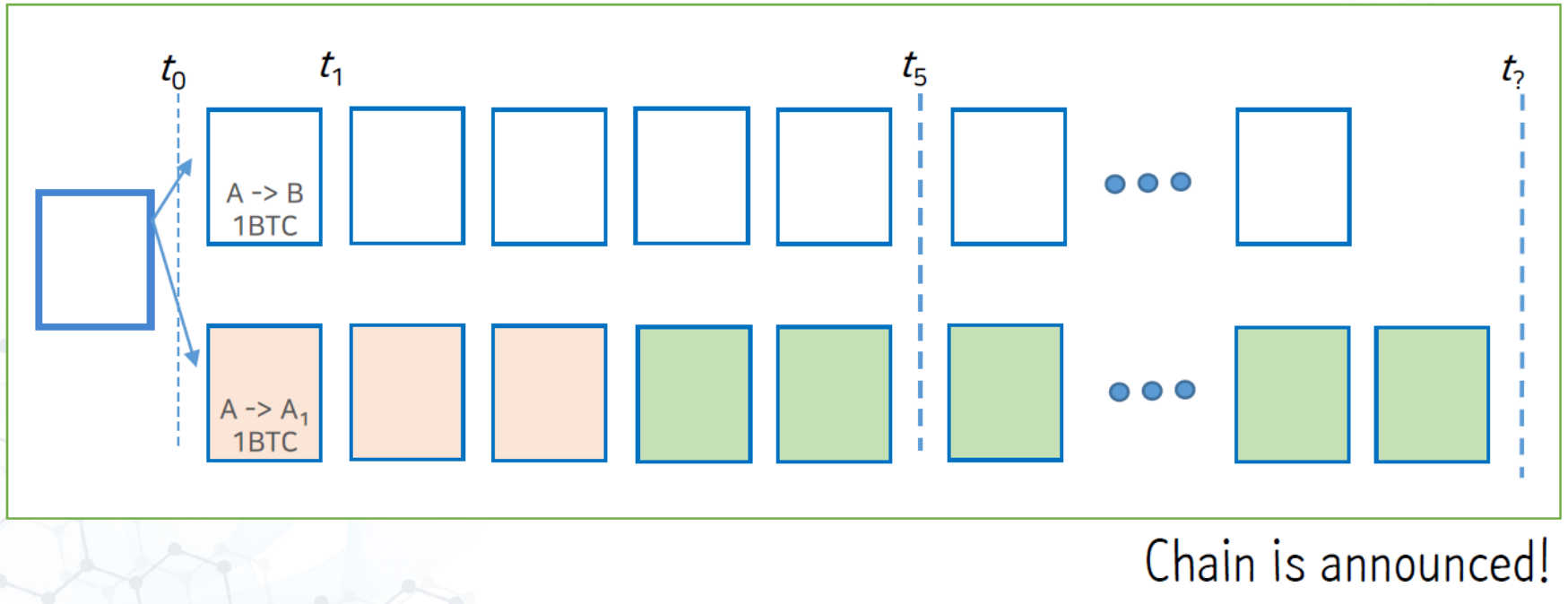
Block을 Chain으로 연결하여 얻는 효과는?



Block을 Chain으로 연결하면 컨텐츠를 못 바꿀까?

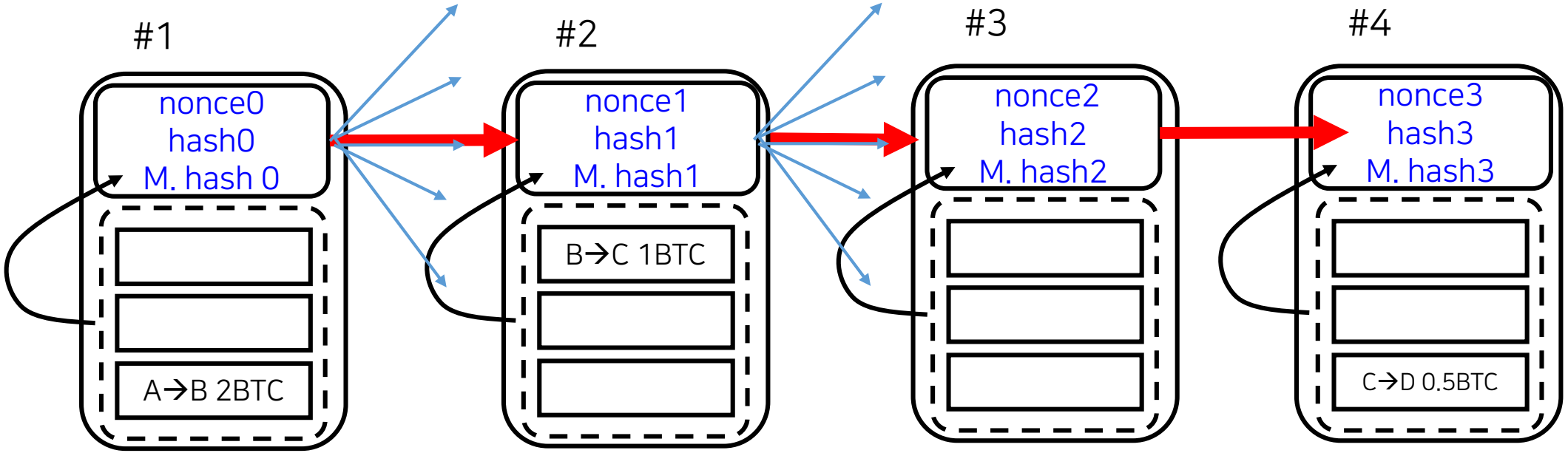


Double Spending이중지불은 어떻게 성립하는가? 방지하려면 무엇이 필요한가?



자세한 내용은 이 링크를 눌러 확인하세요 https://arxiv.org/abs/1903.01711

컨텐츠를 바꿀 수 없게 하려면 무엇이 필요한가?



- Block Chain + PoW
  - Revolutionary new idea!
    - Al-Im-To-Po Theory! https://infonet.gist.ac.kr/?page_id=8954
    - The more come to get involved, the safer the network becomes!
    - Reward
    - Fresh new race for each block.
    - Race means competition.

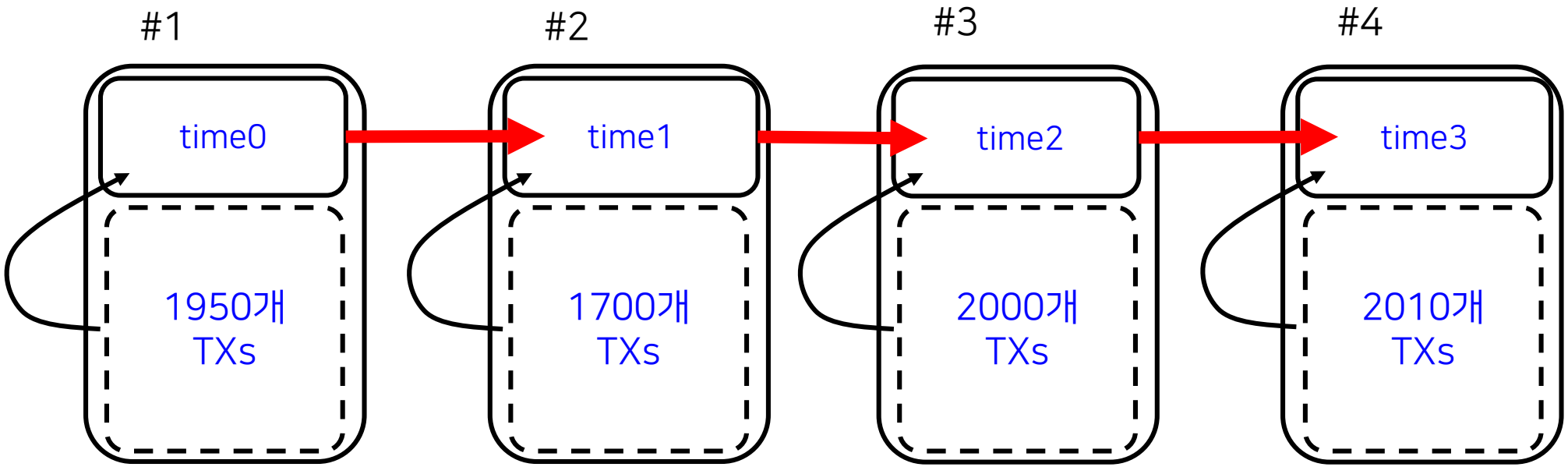
Block #613573

BlockHash 00000000000000000000288ae77f5b8b3c99784f55827431a23f524adf69d0e3a

Summary

Number Of Transactions	3170	**Difficulty**	14776367535688.639
Height	613573 (Mainchain)	**Bits**	17130c78
Block Reward	12.5 BTC	**Size (bytes)**	869316
Timestamp	Jan 20, 2020 1:15:17 AM	**Version**	1073733632
Mined by		**Nonce**	3774951927
Merkle Root	523c52ad89c6ac67bcc01a6...		
Previous Block	613572		

Block Interval, TXs/Block, TPS



TPS = TXs/Block / BI
= 2000 TXs/600초
= 3.3 TXs/sec

TPS제고 필요성 제기

- 인기가 많은 블록체인에만 해당
- TPS를 높이려면?
  1. 블록사이즈를 키우거나
  2. 블록인터벌을 줄이거나
  3. 둘 다 하거나

블록인터벌 **BI** 을 줄이려면?

- 노드 수를 줄이는 프로젝트들이 성행
  - 적은 수의 노드에서 잘 돌아가는 합의알고리즘 사용

  - 그러나 노드 수가 적으면
    - 탈 중앙화는????
    - DS 공격 취약성은???
    - Sybil Attack 취약성은???

블록인터벌 **BI** 을 줄이려면? (2)

- PoW도 Block Interval을 줄일 수 있음
  - Litecoin: 2.5분
  - ETH: Block Interval 14초

  - 수 많은 fork가 매 번 발생하는 문제 발생
    - DS 공격 취약성은???
    - 그러나, Profitable DS Attack분석으로 해결 가능 (See DeSecure블록체인)

PoW Puzzles

- Making PoW puzzles
  - Bitcoin uses SHA256.
  - SHA is *oneway* and *collision free*.
  - Difficulty and Nonce are in the BH.

2 PoW Puzzles

- Finding Good Block Summary
  - Function F takes input x and gives output y :

$$y = F(x)$$

  - x is block header (BH), i.e., F(BH) = hash.
  - Then, it can be written as

F(BH: nonce) < Target PoW Ineq.

  - For a block, find a nonce that satisfies the above inequality (Work)
  - Record the nonce in the block header. (Proof)

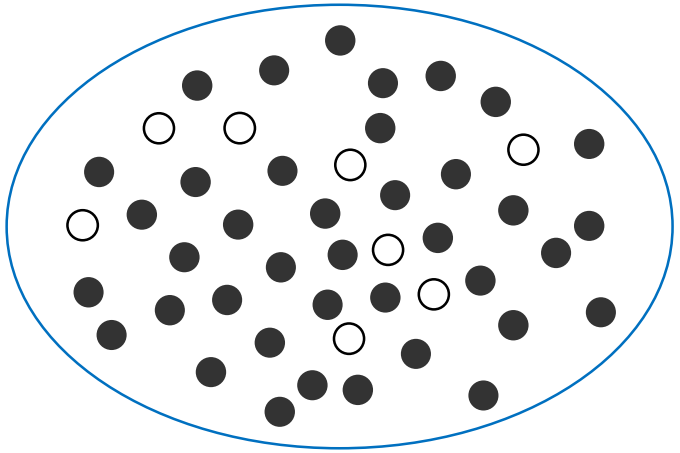
2 PoW Puzzles

- Toy puzzle
  - White and black balls.
  - There are 2^6 balls.
  - Balls are numbers, i.e., hashes.
  - Let Target be 2^3=8.
  - Pick a nonce and run SHA-256.

Total no. of balls $2^6 = 64$
Target = 2^3 0 0 1 0 0 0
White balls = {Balls < Target}
$2^3 - 1 = 7$ 0 0 0 1 1 1
6 0 0 0 1 1 0
5 0 0 0 1 0 1
…

What is the probability that a while ball is picked?

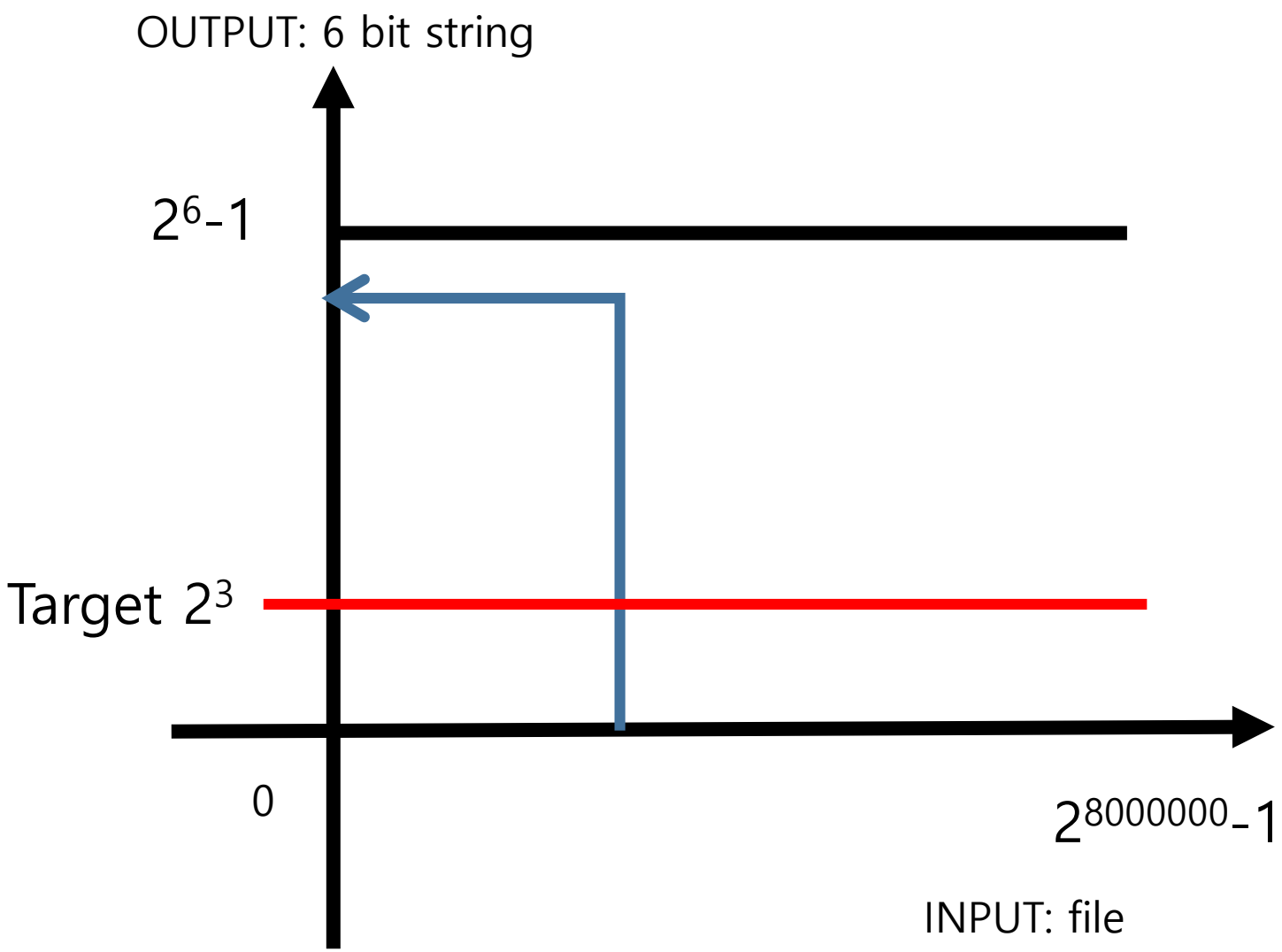
$$p = 2^3/2^6 = 1/8$$



Toy PoW 문제

- SHA-6 사용
- PoW 성공
  - Hash가 8 보다 작은 수가 되는 Input을 찾으면 성공
- 한번 뽑고 PoW 성공 확률?

$$p = 2^3/2^6 = 1/8$$



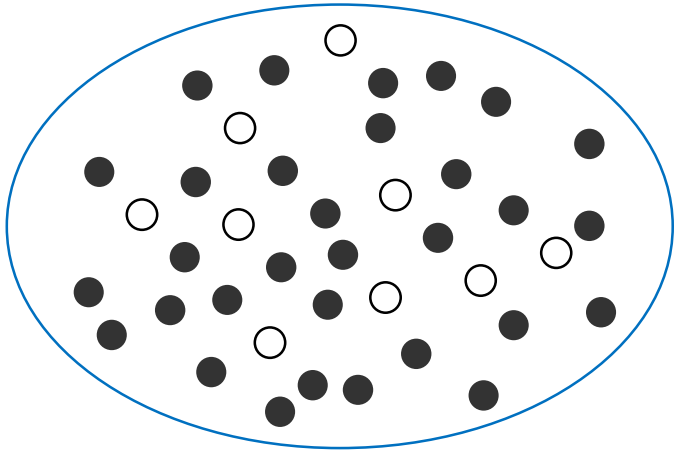
2 PoW Puzzles

- Bitcoin puzzle
  - Hashes are strings of 256 bits.
  - There are 2^{256} hashes in Y.
  - Let Target be $2^{256-16}=2^{240}$.

What is the probability that the hash satisfies the PoW?

$$p = 2^{240}/2^{256}$$
$$= 2^{-16}$$
$$= 1/64000$$

$$Y = \{y | y \text{ is a 256bit string}\}$$



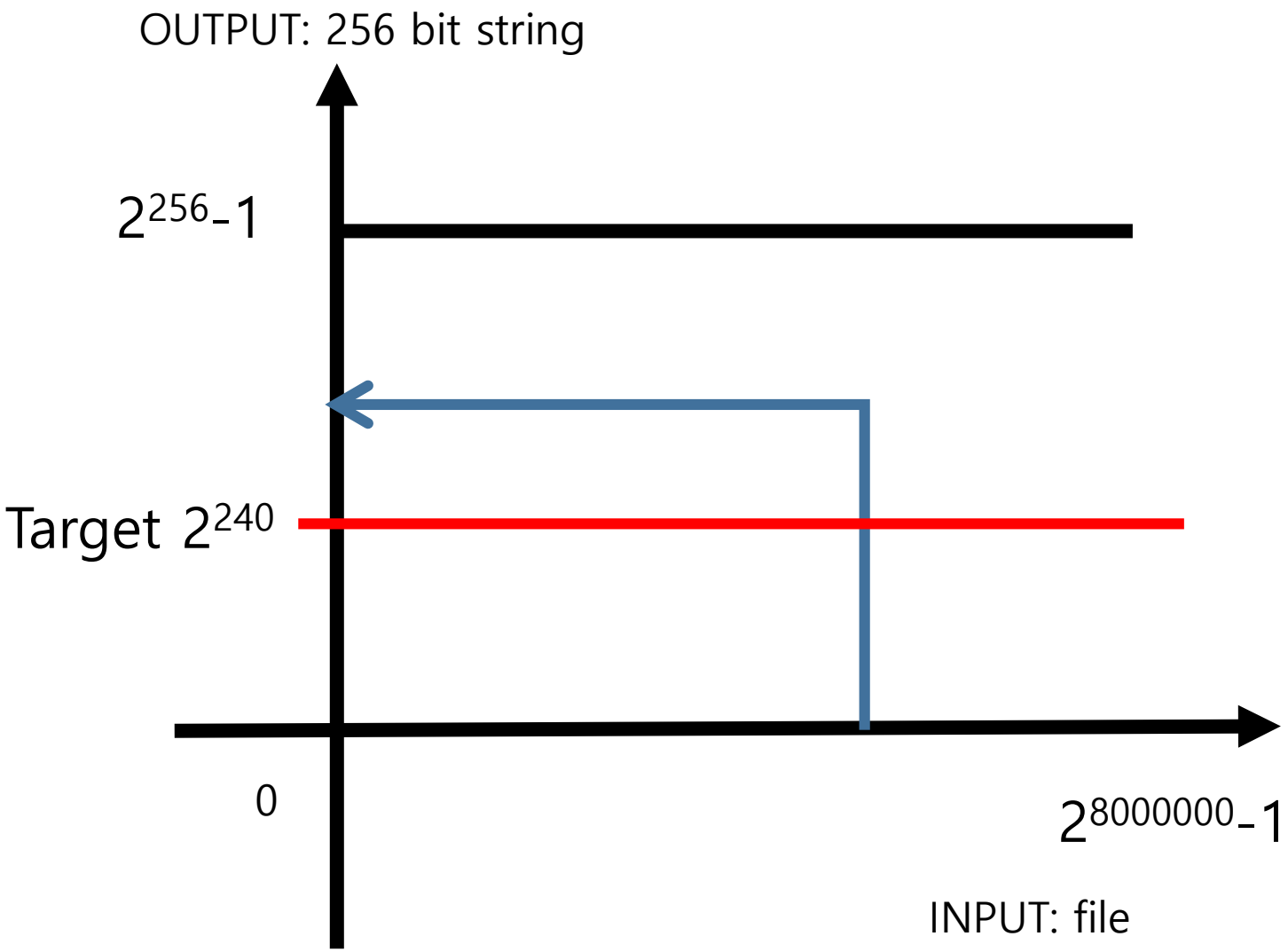
White balls are
64 hexadecimals with 4 leading zeros

"00001642b726b04401627ca9fbac32f5
c8530fb1903cc4db02258717921a4881"

PoW

- SHA-256 사용
- PoW 성공
  - Hash 값이 2^{240} 보다 작은 Input 을 찾으면 성공
- 한번 뽑고 PoW 성공 확률?

$$p = 2^{240}/2^{256}$$
$$= 2^{-16}$$
$$= 1/64000$$



4 Probability of Mining Success

- Given the difficulty p, we aim to find Probability of Mining Success.

- 한 번 뽑을 때 PoW성공 확률이 p 일 때,
  1. k 번째 뽑았을 때 비로서 성공할 확률은?
  2. 평균적으로 몇 번 뽑아야 성공할까?

4 Probability of Mining Success

- (PMF) What is the probability that a CPU solves PoW exactly at the k-th hash?

$$
\begin{aligned}
P_{pmf}(p,\ k) &:= P_p\{K \le k\} - P_p\{K \le k-1\} \\
&= P_p\{K = k\} \\
&= p + (1-p)\,p + (1-p)^2\ p + \cdots + (1-p)^{k-1}\ p \\
&\qquad -\left(p + (1-p)\,p + (1-p)^2\ p + \cdots + (1-p)^{k-2}\ p\right) \\
&= (1-p)^{k-1}\ p \qquad \text{for any } k = 1, 2, 3, \cdots
\end{aligned}
$$

4 Probability of Mining Success

- Average no. of hashes for a PoW success.
  - What is the average number of hashes for a PoW success at a given puzzle difficulty p?

$$\mathbb{E}\{K\} = \sum_{k=1}^{\infty} P_{pmf}(p,\ k)\ k$$

$$= \sum_{k=1}^{\infty} (1-p)^{k}\ p\ k$$

$$= \frac{1}{p}$$

$$= 2^{16} \quad [\text{hashes/block}]$$

Probability of Mining Success

- Definition: Hash Rate of CPU.
  - The hash rate of a CPU is defined as number hashes in a unit time.
  - For example, the hash rate of a CPU which can do 10^6 hash cycles per second is 10^6 hashes/sec.

4 Probability of Mining Success

- ASIC Mining Hardware

Bitcoin Mining Hardware Comparison

Pic	Miner	Hash Power	Price	Buy
	Antminer S9	14.0 TH/s	$3,000	
	Antminer R4	8.6 TH/s	$1,000	

출처: https://www.buybitcoinworldwide.com/mining/hardware/

Bitcoin 난이도 조절

- 평균 **BI**을 10분으로 유지하기 위해 Target값을 조절함

- 문제
- **마이너노드**의 수가 증가하면
  - Network Hash Rate이 낮아/높아 진다.
  - Block Interval이 줄어/길어 진다.
  - 평균 10분 BI을 유지하기 위해서는
  - Target을 크게/작게 해야 한다.

1 Bitcoin Difficulty

- Bitcoin Difficulty (D)

- The aim is to keep the average block generation time be 10 min.
  - Ex) The time span to mine 2016 blocks is set to take 2 weeks.
- Difficulty is adjusted for every 2016 block.
- Measure the time span, T [min], during which the past 2016 blocks were mined.
- Let T_D be 2 weeks [min], i.e., T_D = 2016x10 = 20160 [min].
- If T is different from T_D, adjust the Difficulty D :

$$D = D_{prev} \times \frac{T_D}{T}$$

In Bitcoin, initial D is set to 1 with 8 leading hexa zeros.

- Given a Target, one can determine the network hash rate.
- Suppose you bring your own mining chip.
- You can determine your chance of winning a puzzle.
- It is the ratio of your hash rate to the total hash rate:

$$= \frac{\text{Your Hash Rate}}{\text{Your Hash Rate} + \text{Network Hash Rate}}$$

Bitcoin Difficulty

Ex

- Target 이 2^{204}.
- 채굴에 1 Tera hash/sec mining chip을 구매하여 참여.
- 이때 내가 채굴에 성공할 확률은?
  - The network hash power is $2^{256-\log 2target}/600 = 2^{52}/600 = 7.51e12$ [hash/sec].
  - The hash rate percentage is:

$$= \frac{\text{Your Hash Rate}}{\text{Your Hash Rate} + \text{Network Hash Rate}}$$

$$= \frac{1.00e12}{1.00e12 + 7.51e12}$$

$$= 11.8\%$$

Bitcoin 실전 문제

Bitcoin Difficulty

- 블록 높이 516445 비트코인 블록체인 내 깊이 값 513445에서의 블록들

18 Leading Hexadecimal Zeros

요약	
높이	516445 (Main chain)
해시	00000000000000000004758013a1ed70036479f7d5038c19240afc9fd4710832b
이전 차단	
다음 블록	
시각	2018-04-03 12:40:12
수신 시간	
릴레이된 곳	
난이도	3,511,060,552,899.72
Bits	
거래 수	
출력 합계	
예상된 거래량	816.76804565 BTC
크기	1131.349 KB
번역	0x20000000
Merkle Root	5db080790c0433a7ec8c565932ea75fb7347b6873bc404b2e594f797d7762c10
해시 난수	1225863608
블록 보상	
거래 수수료	

해시 00000000000000000004758013a1ed70036479f7d5036c19240afc9fd4710832b

시간 2018년 4월 3일 12시 40분

난이도 3,511,060,522,899.72 → Log2Diff = 41.68
Target = 256 – (32+41.68) = 256 - 73.68 = $2^{182.32}$

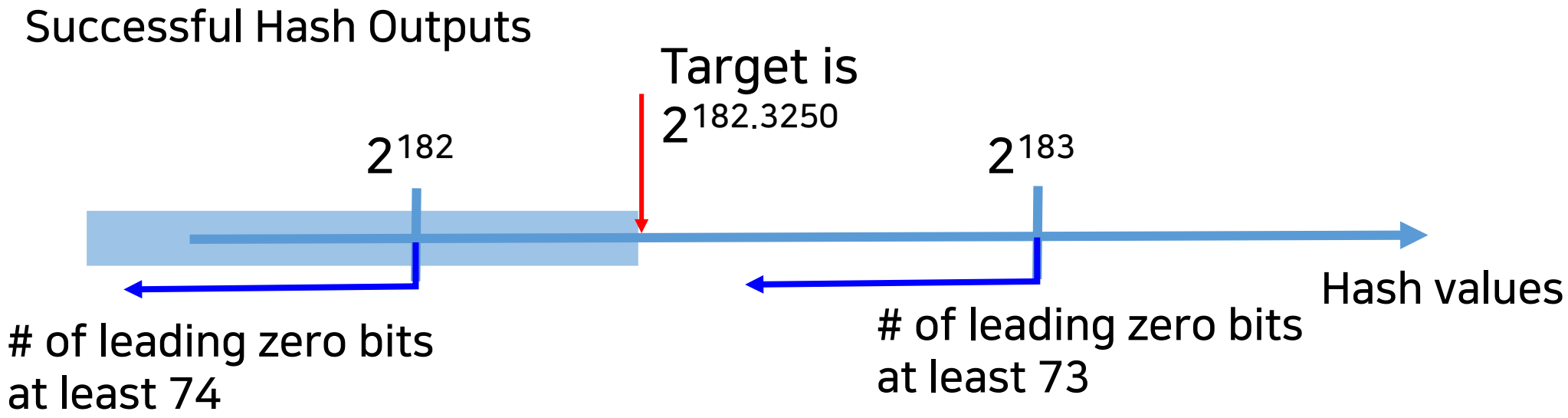
Nonce 1225863608

Bitcoin Difficulty

- Example of Difficulty and Target
  - Block #516445
  - BlockHash 0000 0000 0000 0000 0047 5801 … … … 832b
    - 18 hex zeros * 4 bits/hex + 1 bit = 72 + 1 = 73 zero bits
  - Difficulty D is 3,511,060,552,899.7197 = 3.5e12
  - New Target is $Target_0$ * (1/D)
    - Log2(D) = 41.68
    - Target = $2^{224.000}$ $2^{-41.675}$
      = $2^{182.325}$

Bitcoin Difficulty

Recall PoW Success
is
SHA Hash Output < Target



Bitcoin Difficulty

- Network Hash Rate : Block#516445
- With D=3.5e12, the probability p is about $2^{-(32+41.675)} = 2^{-73.6750}$.
- Then, it would take $1/p = 2^{73.6750} \sim$ 1.5080 e22 hashes to mine a single block.
- Dividing it by 10 min = 600 sec, the network hash rate is obtained, 25.13 Exa hash/sec.

Exa = 10^{18}

10% 해쉬 파워 확보를 위해 투자해야 할 돈은?

Ex

- Antminer S9s (14 Thps) 이 몇 개 필요할까, 만약 내가 hash power를 0.01 % 갖기를 원한다면?
- 네트워크 hash rate 은 25 Exa hash/sec 이다.

  - You need to bring at least 179 AS9 chips.

$$\text{Your Hash Rate} \geq \frac{0.01\% \;\; \text{Network Hash Rate}}{100\% - 0.01\%}$$

$$= \frac{1}{9999} 25e18 = 25e14$$

$$= 178.6(14e12)$$

- 179000개
- 단가 3000 USD
- 537 MUSD

문제

네트워크 hash rate은 25 Exa hash/sec 이다.

- 현재 네트워크에서 작동하는 마이닝 노드의 최소 개수는 몇 개인가?

답

네트워크 hash rate은 25 Exa hash/sec 이다.

- 현재 비트코인 네트워크에서 작동하는 마이닝 노드의 최소 개수는 몇 개인가?

$$\text{최소갯수} = \frac{\text{Net HR}}{\text{가장빠른칩 HR}}$$

$$= \frac{25 \times 10^{18}}{14 \times 10^{12}}$$

$$\sim 1.8 \times 10^{6}$$

문제

네트워크 hash rate은 25 Exa hash/sec 이다.

- 한개의 가장 빠른 마이닝칩으로 하나의 블록 작업증명에 성공하려면 걸리는 시간은?

문제

네트워크 hash rate은 25 Exa hash/sec 이다.

- 한개의 가장 빠른 마이닝칩으로 하나의 블록 작업증명에 성공하려면 걸리는 시간은?

$$\text{시간} = \frac{\text{Net HR} \times 600\text{초}}{\text{가장빠른칩 HR}}$$

$$= \frac{25 \times 10^{18}}{14 \times 10^{12}} \times 600\text{초}$$

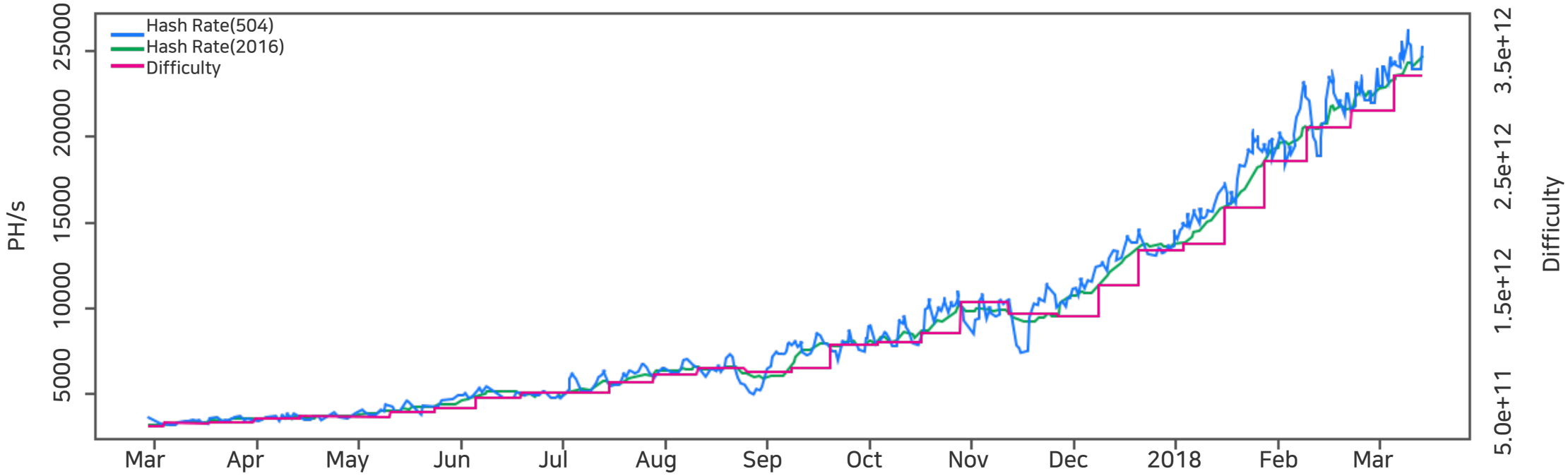
$$\sim 1.07 \times 10^{9}\text{초}$$

$$= 31.7 \text{ 년}$$

일년 = 3153만6천 초

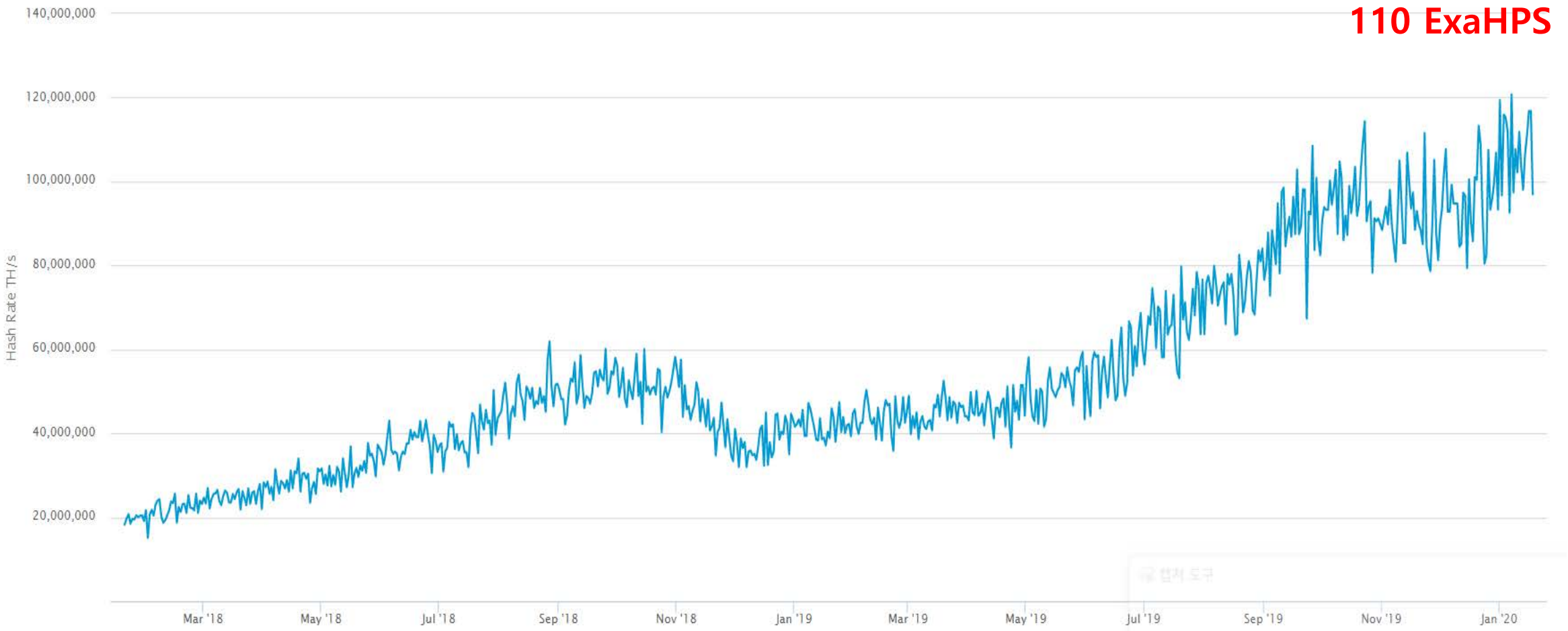
2 History of Bitcoin difficulty

- Bitcoin Hash Rate vs Difficulty (Mar/17 ~ Apr 18)



출처: https://bitcoinwisdom.com/bitcoin/difficulty

Network Hash Rate = 1/p



비트코인 PoW가 좋은 점

- PoW provides flexibility in TPS solutions!
  - Small difficulty → fast TPS
  - Large difficulty → slow TPS

- PoW is sufficient for data immutability!

- PoW and blockchain is a technological breakthrough.

PoW는 근본적인 보안성 해결방안

- 탈 중앙성과 보안성 모두 확보 가능한 해결방안!

- 주요 합의알고리즘은 탈 중앙성을 희생하여 보안성과 확장성을 얻으려 목표, 혹은 off-chain 정책에 의존함.
  - Staking, Delegation, Activity-checking, Leader Selecting, Random Selecting, etc.

- PoW의 현재 문제점을 새로운 기술로 해결하면 여전히 가장 좋은 해결방안이 됨!

블록생성과 검증이 분리되면 어떤 일이?

- 랜덤으로 선출된 자가, 지분을 많이 가진 자가, 활동이 많은 자가, 장군으로 선출된 자가, …

  블록을 생성하세요.

- 나머지는 검증을 하겠습니다.
- 선출 → 생성 → 검증 등 여러 단계로 분리 됨.
- 그러나, 각 단계에서 일이 제대로 되었을까?
  - Validators를 뽑아서 합시다. 보상은 나눕시다.
- 생성자 혹은 검증자가 일을 대충해서 좋지 않은 TX가 들어갈 경우?
  - 각 단계에서 혹시 실패한 경우가 발생할 경우는…
  - 평판도 측정, Staking 압수, …

PoW는 생성과 검증이 동시에 이루어짐

- 수 많은 노드가 생성과 검증을 경쟁적으로 (동시에) 함
- 끊김이 없이 블록이 생성됨
- 앞으로 전진만 있음
- 잘못된 TX이 포함될 경우 헛고생, 스스로 검증하고 생성함

Goal of this lecture note

- Bitcoin Script
- Tables of OP Codes
- Easy Script
- Pay-to-Public Key Hash (P2PKH) Script
- Multisignature and Smart Contracts Scripts

Bitcoin Script

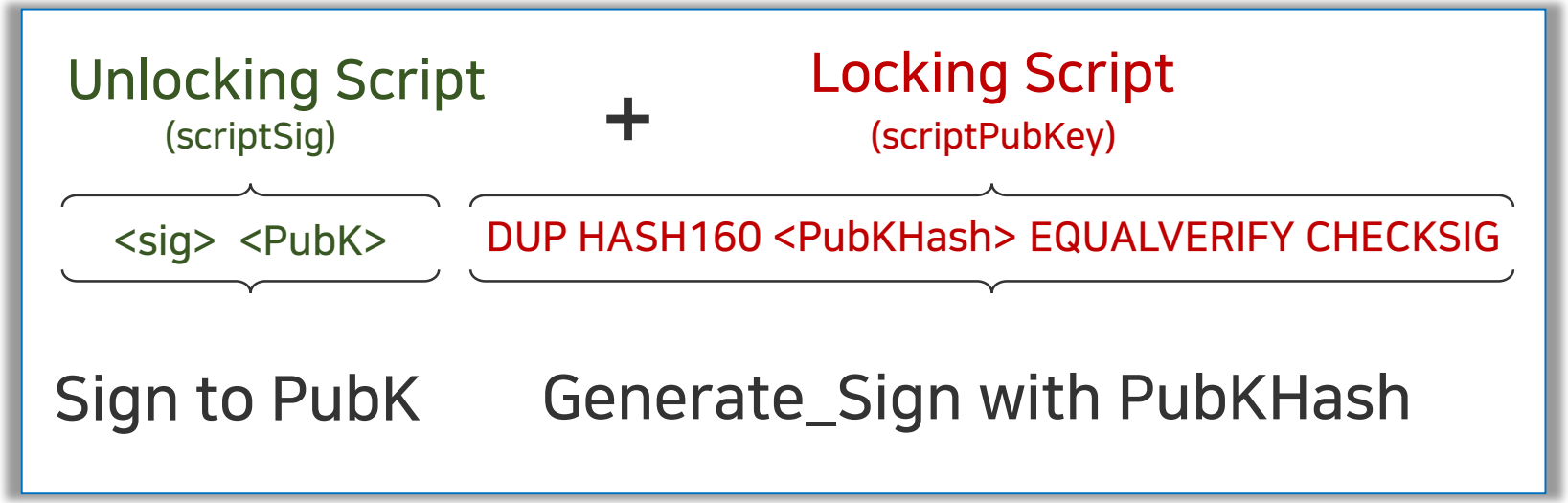
- Bitcoin Script
  - Bitcoin uses a scripting language for transactions.
  - A script is simple, stack-based, and processed from left to right.
  - It is intentionally not Turing-complete, with no loops.
  - A script is a list of instructions.
  - The payer locks the vout value to a payee's public address.
  - The payee unlocks the lock by providing the signature.

- Bitcoin Script
  - Payer uses a lock script to lock the `vout` value to a destination Bitcoin address and payee uses an unlock script to spend it.
    1. The `vout` value transferred to a destination address mapped from a public key is locked into the locking script, and
    2. A signature is embedded in the unlocking script which proves the ownership of the private key corresponding to the locked value.
  - Further reading from https://en.bitcoin.it/wiki/Script

- See if scriptSig unlocks scriptPubKey!
  - *Script Construction (Unlock+Lock)*
    - The locking script is called a *scriptPubKey*, because it contains a public key or a Bitcoin address.
    - The unlocking script is called *scriptSig* because it contains a digital signature.
    - When a correct unlocking script is provided to the locking script, the execution of the complete script comes out TRUE.
    - Then, the provider of scriptSig can spend the value.

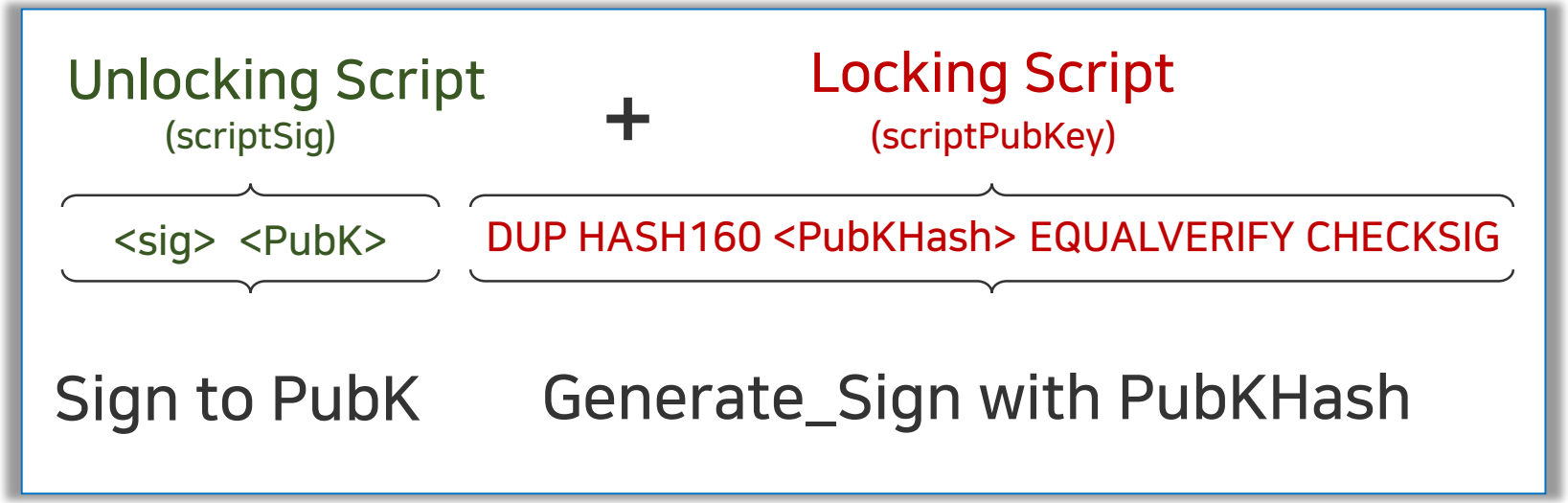
Bitcoin Script

- Pay to Public Key Hash
  - 시간 1: A's Sign (Priv. Key)→ Lock to Pub. Key of B 2.0BTC.
  - 시간 2: B's Sign (Priv. Key)→ Lock to Pub. Key of C 1.0BTC.
  - 시간 3: C's Sign (Priv. Key)→ Lock to Pub. Key of D 0.5BTC.



Bitcoin Script

- Values provided by users are given in < >.
- DUP, HASH160, EQUALVERIFY, CHECKSIG are Operations.



2 Tables of OP Codes

- Table C-7. Cryptographic and Hashing Operations

Symbol	Value(hex)	Description
OP_RIPEMD160	0xa6	Return RIPEMD160 hash of top item
OP_SHA1	0xa7	Return SHA1 hash of top item
OP_SHA256	0xa8	Return SHA256 hash of top item
OP_HASH160	0xa9	Return RIPEMD160(SHA256(x)) hash of top item
OP_HASH256	0xaa	Return SHA256(SHA256(x)) hash of top item
OP_CODESEPARATOR	0xab	Mark the beginning of signature-checked data

Appendix C of Mastering Bitcoin

2 Tables of OP Codes

- Table C-7. Cryptographic and Hashing Operations

Symbol	Value(hex)	Description
OP_CHECKSIG	0xac	Pop a public key and signature and validate the signature for the transaction's hashed data, return TRUE if matching
OP_CHECKSIGVERIFY	0xad	Same as CHECKSIG, then OP_VERIFY to halt if not TRUE
OP_CHECKMULTISIG	0xae	Run CHECKSIG for each pair of signature and public key provided. All must match. Bug in implementation pops an extra value, prefix with OP_NOP as workaround
OP_CHECKMULTISIGVERIFY	0xaf	Same as CHECKMULTISIG, then OP_VERIFY to halt if not TRUE

2 Tables of OP Codes

- Table C-3. Stack Operations

Symbol	Value(hex)	Description
OP_TOALTSTACK	0x6b	Pop top item from stack and push to alternative stack
OP_FROMALTSTACK	0x6c	Pop top item from alternative stack and push to stack
OP_2DROP	0x6d	Pop top two stack items
OP_2DUP	0x6e	Duplicate top two shack items
OP_3DUP	0x6f	Duplicate top three shack items
OP_2OVER	0x70	Copies the third and fourth items in the stack to the top
OP_2ROT	0x71	Moves the fifth and sixth items in the stack to the top
OP_2SWAP	0x72	Swap the two top pairs of items in the stack
OP_IFDUP	0x73	Duplicate the top item in the stack if it is not 0
OP_DEPTH	0x74	Count the items on the stack and push the resulting count

Appendix C of Mastering Bitcoin

2 Tables of OP Codes

- Table C-3. Stack Operations

Symbol	Value(hex)	Description
OP_DROP	0x75	Pop the top item in the stack
OP_DUP	0x76	Duplicate the top item in the stack
OP_NIP	0x77	Pop the second item in the stack
OP_OVER	0x78	Copy the second item in the stack and push it on to the top
OP_PICK	0x73	Pop value N from top, then copy the Nth item to the top of the stack
OP_ROLL	0x7a	Pop value N from top, then move the Nth item to the top of the stack
OP_ROT	0x7b	Rotate the top three items in the stack
OP_SWAP	0x7c	Swap the top three items in the stack
OP_TUCK	0x7d	Copy the top item and insert it between the top and second item

2 Tables of OP Codes

- Table C-6. Numeric Operators

Symbol	Value(hex)	Description
OP_1ADD	0x8b	Add 1 to the top item
OP_1SUB	0x8c	Subtract 1 from the top item
OP_2MUL	0x8d	Disabled (Multiply top item by 2)
OP_2DIV	0x8e	Disabled (Divide top item by 2)
OP_MEGATE	0x8f	Flip the sign of top item
OP_ABS	0x90	Change the sign of the top item to positive
OP_NOT	0x91	If top item is 0 or 1 boolean flip it, otherwise return 0
OP_ONOTEQUAL	0x92	If top item is 0 return 0, otherwise return 1
OP_ADD	0x93	Pop top two items, add them and push result

Appendix C of Mastering Bitcoin

2 Tables of OP Codes

- Table C-2. Conditional Flow Control

Symbol	Value(hex)	Description
OP_NOP	0x61	Do nothing
OP_VER	0x62	Halt – Invalid transaction unless found in an unexecuted OP-IF clause
OP_IF	0x63	Execute the statements following if top of stack is not 0
OP_NOTIF	0x64	Execute the statements following if top of stack is 0
OP_VERIF	0x65	Halt – Invalid transaction
OP_VERMPTIF	0x66	Halt – Invalid transaction
OP_ELSE	0x67	Execute only if the previous statements were not executed
OP_ENDIF	0x68	Ends the OP_IF, OP_NOTIF, OP_ELSE block
OP_VERIFY	0x69	Check the top of the stack, Halt and Invalidate transaction if not TRUE

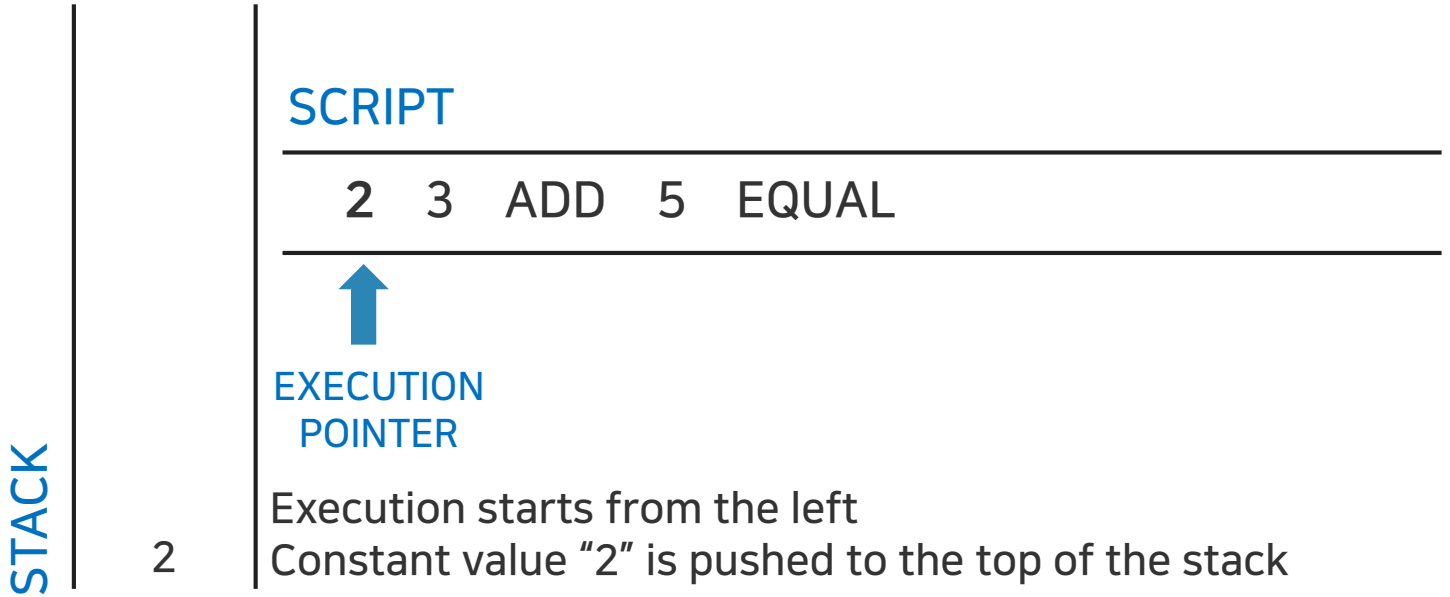
Appendix C of Mastering Bitcoin

Bitcoin opcodes

- Opcode가 뭔가요?
  - 일반 컴퓨터에서 실행되는 기본 프로그램
- 몇 개인가요?
  - 최 대 16 x 16, 즉 256개

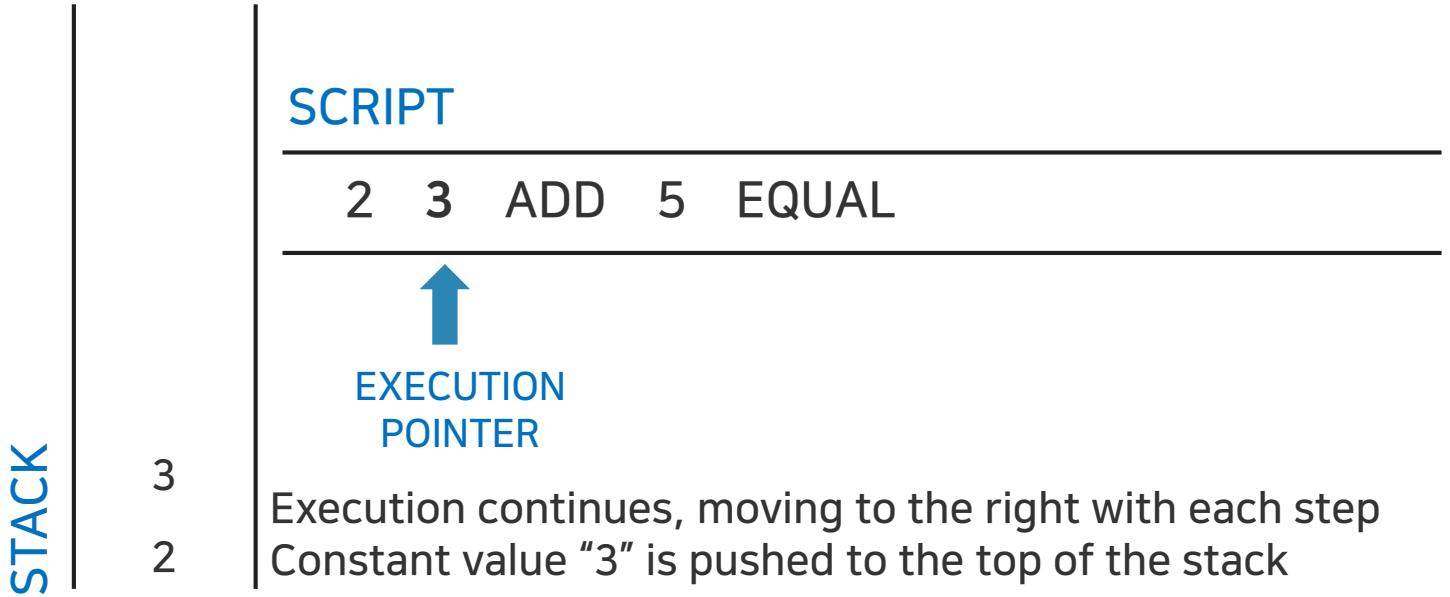
3 Easy Script

- Example script: 2 + 3 = 5



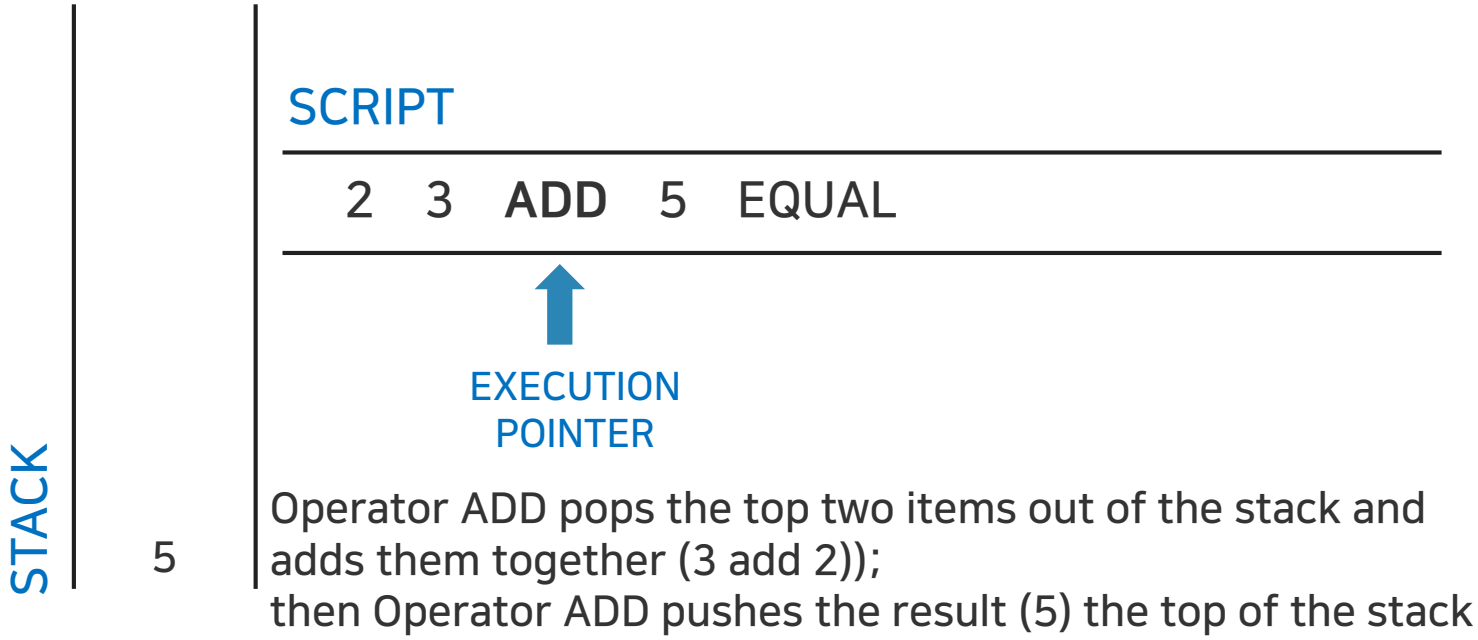
Easy Script

- Example script: 2 + 3 = 5



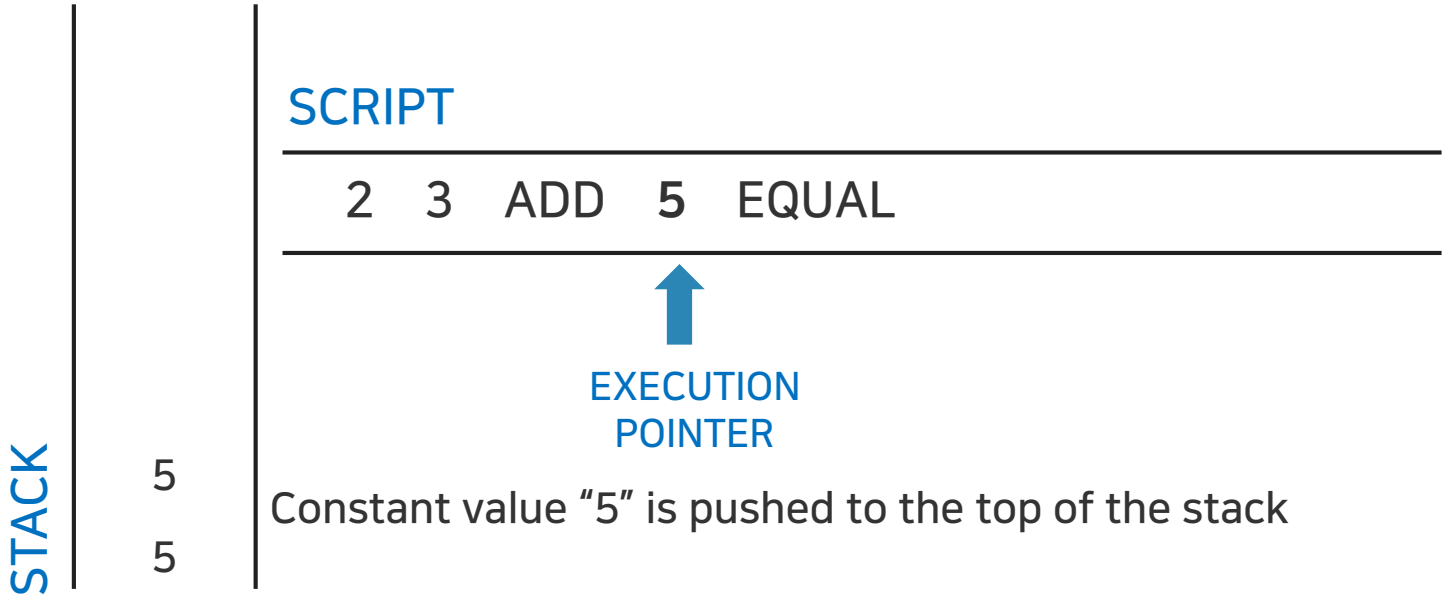
3 Easy Script

- Example script: 2 + 3 = 5



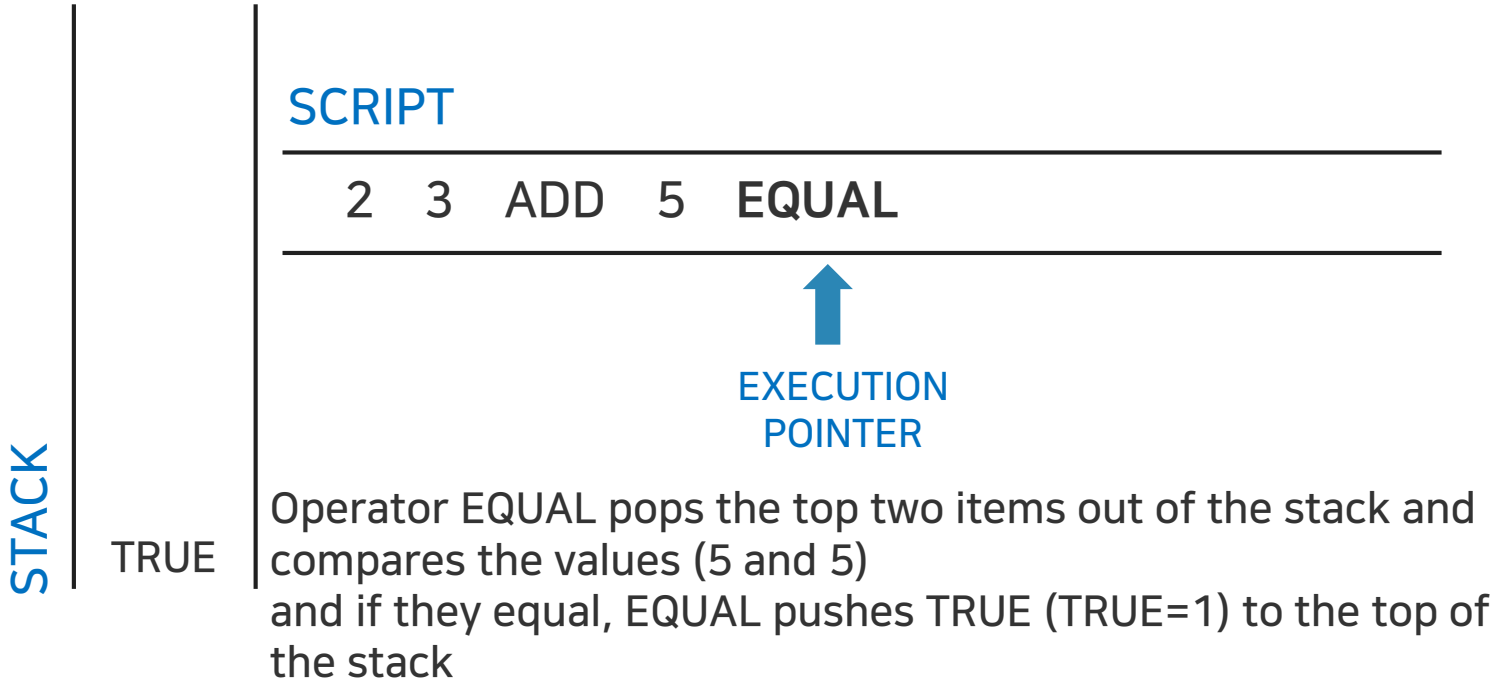
Easy Script

- Example script: 2 + 3 = 5



Easy Script

- Example script: 2 + 3 = 5



- Unlock + Lock Pair, shows a proof of ownership
  - Use a part of the arithmetic example script as the locking script:

    ```
    3 OP_ADD 5 OP_EQUAL
    ```

  - Which can be satisfied by a transaction containing an input with the unlocking script:

    ```
    2
    ```

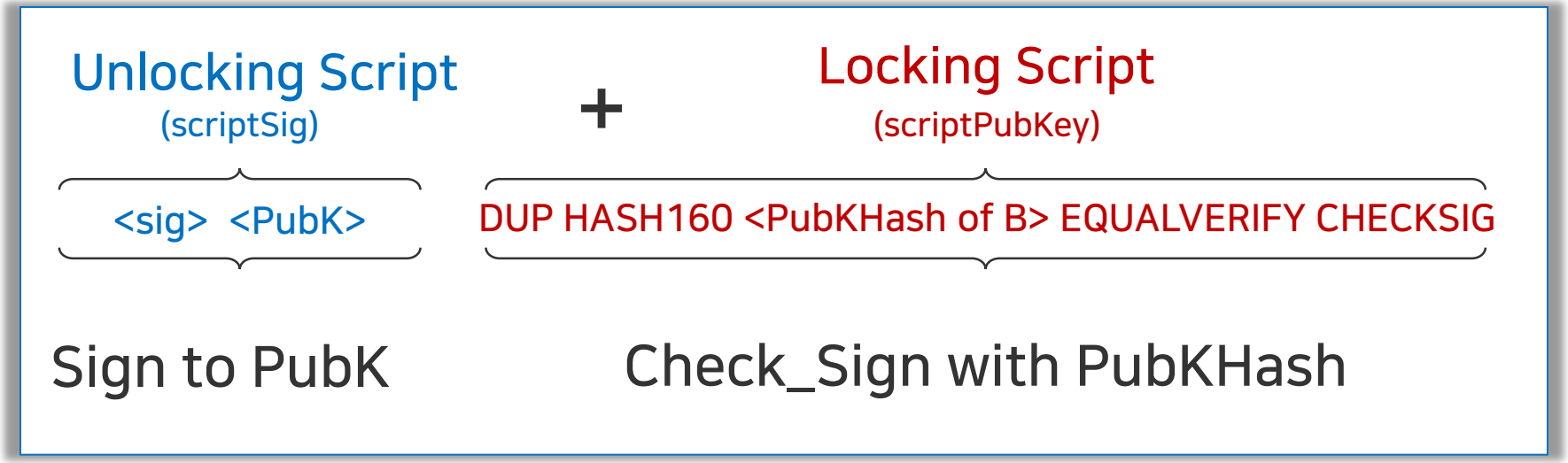
  - Put them together, we have the complete script.

    ```
    2 3 OP_ADD 5 OP_EQUAL
    ```

  - This pair will produce an outcome of TRUE.

4 P2PKH Script

- Now let us make a more realistic pair focusing on B.
  - 시간 1: A's Sign (Priv. Key)→ Lock to Pub. Key of B 2.0BTC.
  - 시간 2: B's Sign (Priv. Key)→ Lock to Pub. Key of C 1.0BTC.
  - the signature.



- P2PKH of B
  - Unspent value belongs to Pay to Public Key Hash(P2PKH) script.

```
OP_DUP OP_HASH160 <Public Key Hash of B> OP_EQUAL OP_CHECKSIG
```

  - Unlocking script is a digital sign created by corresponding private key.

```
<sig of B>  <PubK of B>
```

- Locking script with a single <input>
  - One input, four operations
  - `OP_DUP`: duplicate
  - `OP_Hash160`(x)= `RIPEMD`(`SHA256`(x))
  - `<Public Key Hash of B>`
  - `OP_EQUAL`: return TRUE if the two top most values are equal
  - `OP_CHECKSIG`: checks to see if the provided `sign` and `pubkey` are valid

4 P2PKH Script

- Locking script with <input>

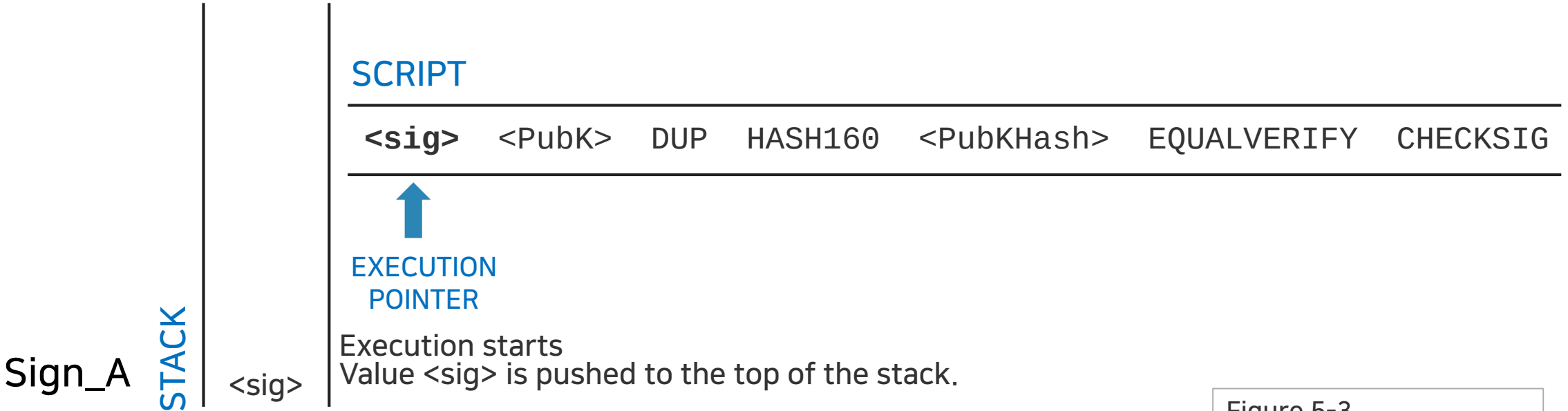


Figure 5-3.
Evaluating a script for a Pay-to-Public-Key-Hash transaction (Part 1 of 2)

4 P2PKH Script

- Locking script with <input>

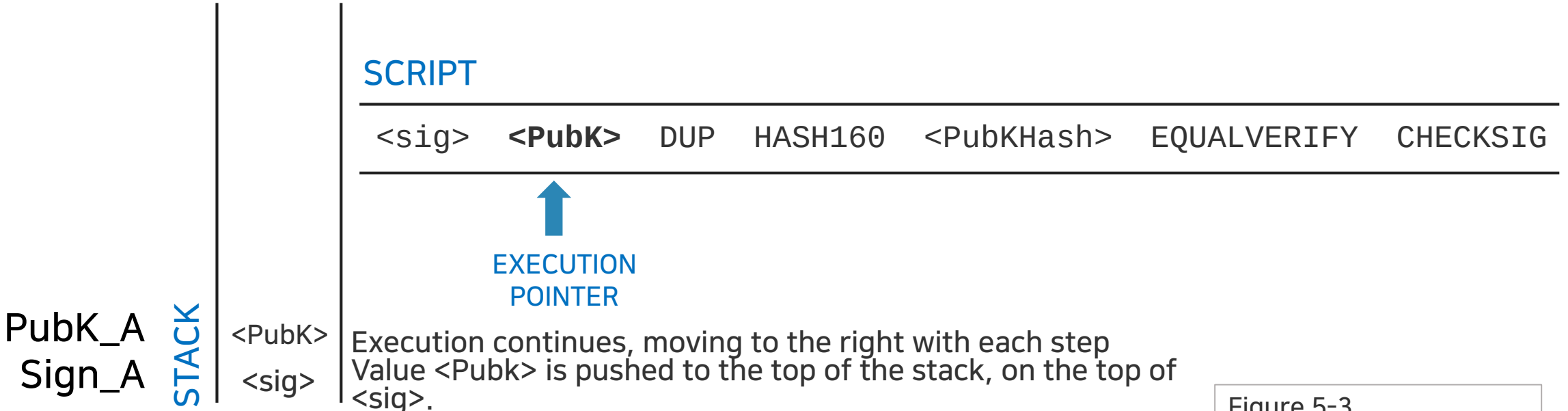


Figure 5-3.
Evaluating a script for a Pay-to-Public-Key-Hash transaction (Part 1 of 2)

4 P2PKH Script

- Locking script with <input>

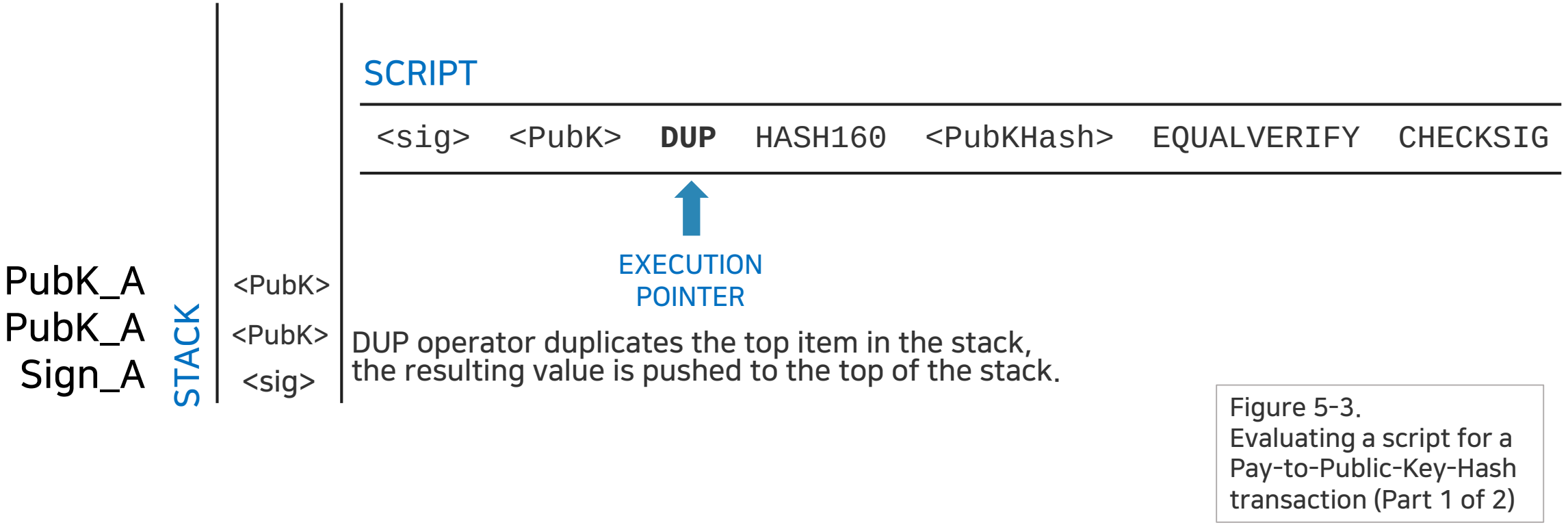


Figure 5-3.
Evaluating a script for a Pay-to-Public-Key-Hash transaction (Part 1 of 2)

4 P2PKH Script

- See if two PubKH_As match

PubKH_A
PubK_A
Sign_A

STACK

<PubK Hash>
<PubK>
<sig>

SCRIPT

<sig> <PubK> DUP **HASH160** <PubKHash> EQUALVERIFY CHECKSIG

EXECUTION POINTER

HASH160 operator hashes the top item in the stack with RIPEMD160(SHA256(PubK)).
the resulting value (PubKHash) is pushed to the top of the stack.

4 P2PKH Script

- See if the two PubKH_As match

PubKH_A
= ab3813c
PubKH_A
PubK_A
Sign_A

STACK

<PubK Hash>
<PubK Hash>
<PubK>
<sig>

SCRIPT

<sig> <PubK> DUP HASH160 **<PubKHash>** EQUALVERIFY CHECKSIG

EXECUTION POINTER

The value PubKHash from the script is pushed on top of the value PubKHash calculated previously from the HASH160 of the PubK.

4 P2PKH Script

- Check Signature

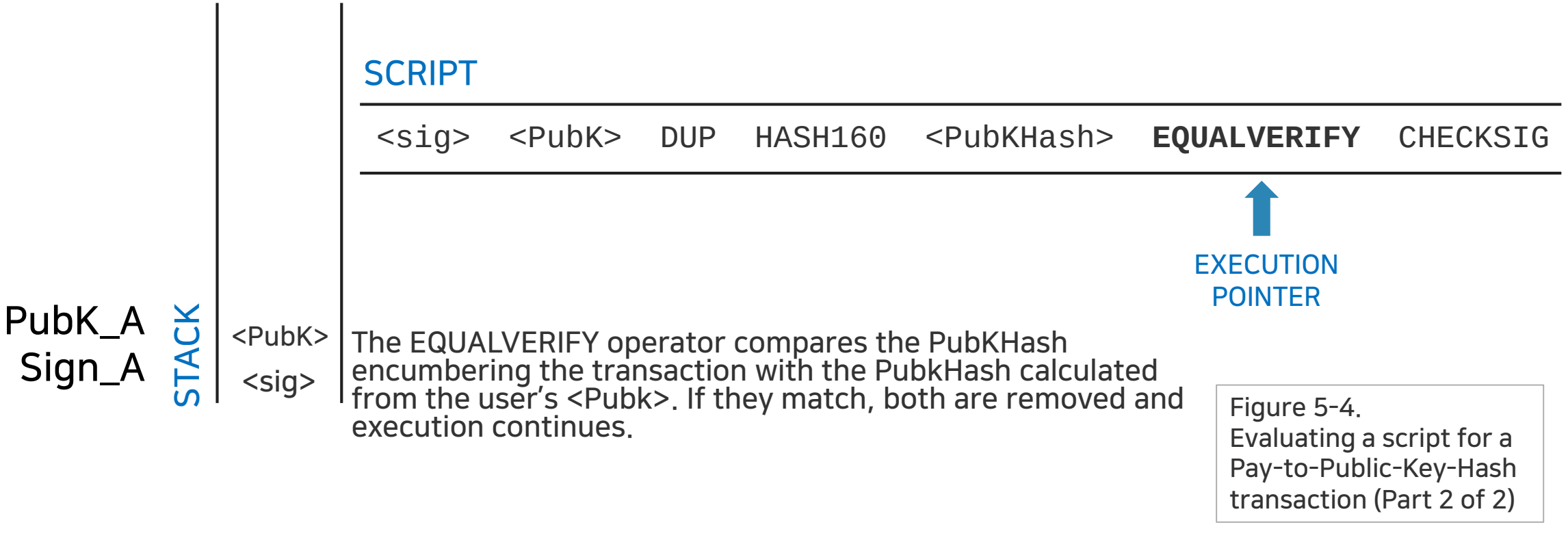


Figure 5-4.
Evaluating a script for a Pay-to-Public-Key-Hash transaction (Part 2 of 2)

- Recall `SignGenerate` and `isSignatureValid` routines
  - m = {`TXID`, output [n] = {value, a locking script with `PKH_A`}}
  - `Sign_A` = `SignGenerate` (m, `k_A`);
  - `isSignatureValid`(m, `Sign_A`, `PK_A`) = `TRUE/False`

P2PKH Script

- Check Signature

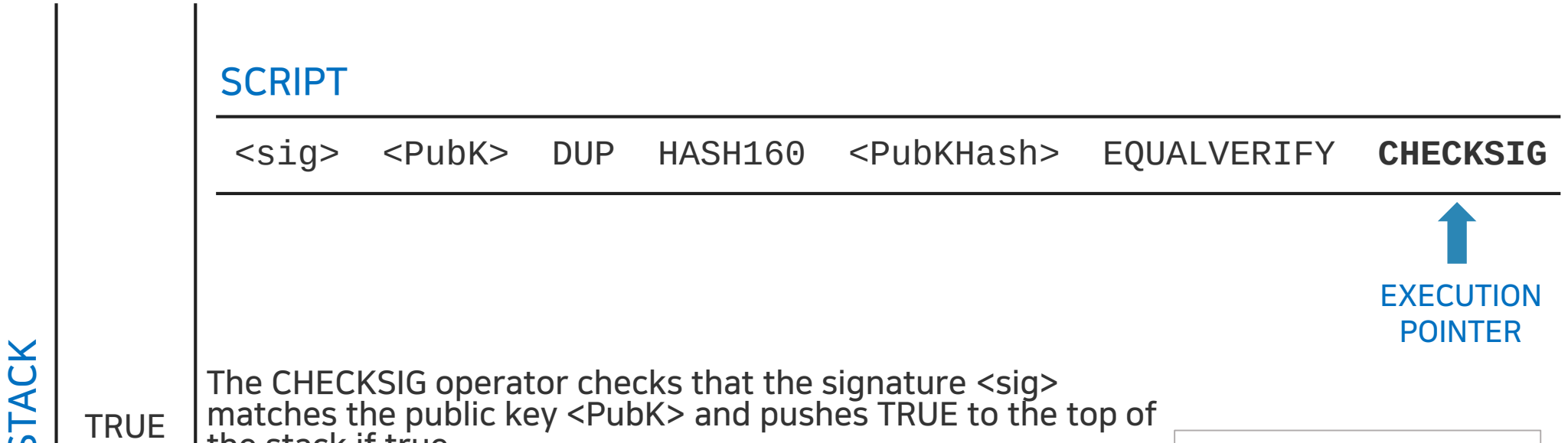


Figure 5-4.
Evaluating a script for a Pay-to-Public-Key-Hash transaction (Part 2 of 2)

Multisignature and Smart Contracts Scripts

- Other Scripts
  - Pay to Public Key (P2PK), introduced in the Bitcoin white paper.
  - Pay to Public Key Hash (P2PKH), used in the code by Satoshi Nakamoto.
  - Pay to Script Hash (P2SH), introduced winter of 2012.
    - These Bitcoin addresses are beginning with 3.
    - Hash of a script is the beneficiary.
    - It can be used for a `multisignature` script.
      - M out of N keys are needed to spend the value.
      - Useful for joint accounts

Multisignature and Smart Contracts Scripts

- Bitcoin uses scripts for Smart Contracts
  - There are many different possibilities that can be expressed with this scripting language.
  - Smart contracts can be programmed in to code which expresses more complex conditions for spending and how these conditions can be satisfied by unlocking scripts.
  - *This language allows for a nearly infinite variety of conditions to be expressed.*
  - *This is how bitcoin gets the power of "programmable money." (Mastering Bitcoin)*

Multisignature and Smart Contracts Scripts

- Bitcoin does not allow any loop for stable operations.
- Ethereum does.
  - `Jump` and `JumpTo` are used in the list of OP codes.
  - https://github.com/crytic/evm-opcodes.
- Bitcoin is more prudent and focuses on safety.

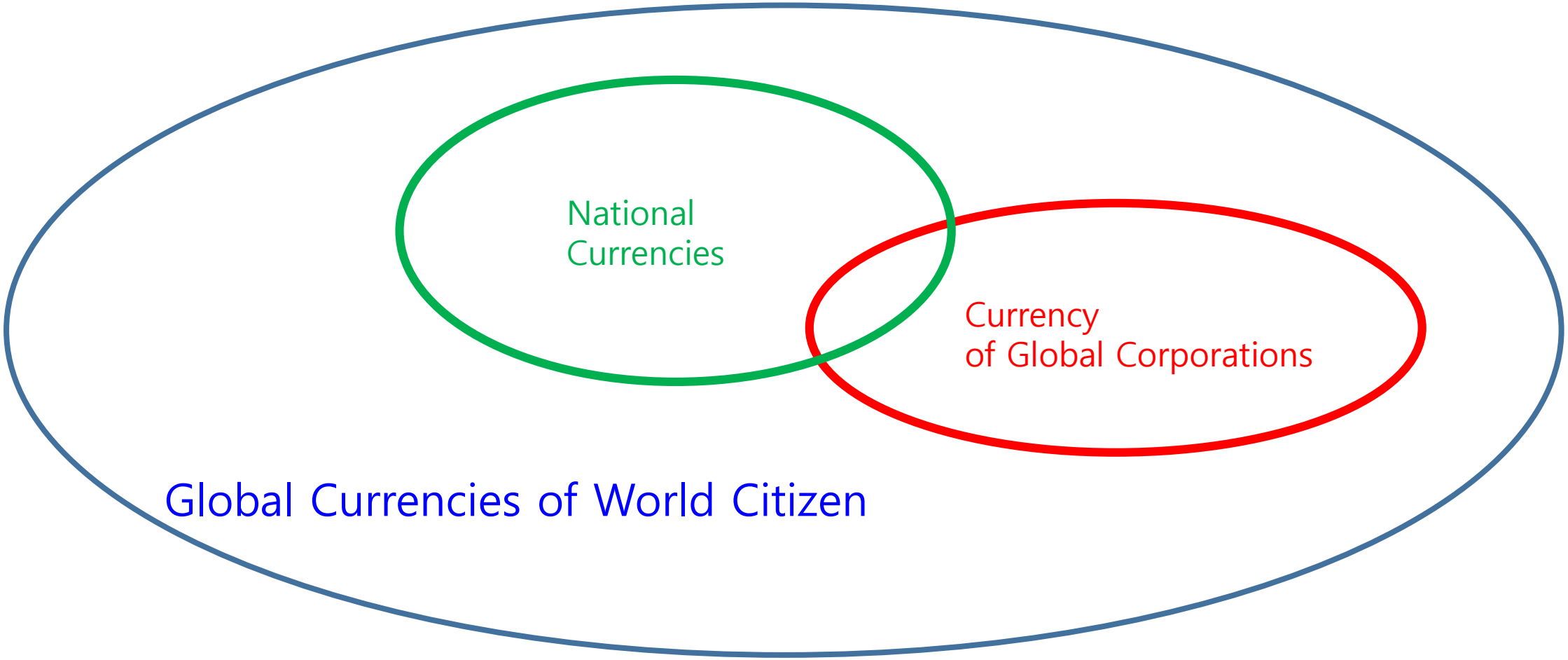
Bitcoin is money of people!

- We are people from the globe.
- We have our computers shared.
- This computer network maintains the book of transactions.
- This computer network mints coins.
- Currency is created by us the people.
- We the people get the seigniorage.
- Supply of this global currency is fixed.

Bitcoin provides honest money!

- It is a money freely traded worldwide.
- Supply is fixed.
- Participation is open.
- BIT-ECC is more decentralized than BIT.

10 years from now, competing currencies



Concluding Remarks

- Bitcoin is the currency of world citizen.
- PoW is fundamental to decentralization and security.
- Centralized altcoins are not as valuable as Bitcoin.
- DeSecure blockchains and multi-level blockchains provide new technological advance.
- It is inevitable to use cryptocurrencies.

References

- https://gist.edwith.org/bitcoin-tech Starmooc 동영상 강의 총 14 차시

- Heung-No Lee, “DeSecure Blockchains,” ETH-CON, Seoul, Reb. of Korea., May 2019.

- Heung-No Lee, “Blockchain Consensus and Governance,” July Meet-Up, Institute of Blockchain and Law, July 11th, 2019. YouTube Video Available at https://www.youtube.com/watch?v=7ujkFgsKPdY.

Selected References of GIST Blockchain Economy Center

- [Lee1] JH Jang and Heung-No Lee, "Profitable Double Spending Attacks," March 5th, 2019 submitted to IEEE Trans. Information Forensics and Securities, downloadable from https://arxiv.org/abs/1903.01711.
- [Lee2] 장재혁, 이흥노, "50%미만 이중 지불 공격", OSIA S&TR Journal, Vol. 32, No. 1, Mar. 2019. (pdf)
- [Lee3] 정현준, 이흥노, "암호화폐 투자와 규제 현황", 한국정보과학회, 정보과학회지, 제 36권, 제 12호, pp. 49-56, Dec, 2018. (pdf)
- [Lee4] 박상준, 김형성, 이흥노, "Introduction to Error-Correction Codes Proof of Work," 블록체인경제 특집호, 대한전자공학회지, June 2019.
- [Lee5] Sangjun Park, HS Kim, Heung-No Lee, "Time-Variant Proof-of-Work Using Error-Correction Codes," to be submitted to IEEE Trans. Information Forensics and Securities.
- [Lee6] Mohamed Yaseen.J, Giljun Jung and Heung-No Lee."Decentralized Framework for Medical Images Based on Blockchain and Inter Planetary File System", The 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society(EMBC 2019), Berlin, Germany, Jul. 23-27, 2019.
- [Lee7] Please visit INFONET home page https://infonet.gist.ac.kr/?page_id=14 for more references.