블록체인 기술과 응용 서비스

1. 블록체인 개요
2. 블록체인 응용과 사례
3. 블록체인 보안
4. **비트코인 블록체인의 구조와 동작원리**
5. 이더리움 블록체인의 구조와 동작원리
6. 가상전자 거래소, 전자지갑, 채굴
7. 블록체인 이슈와 전망

대구사이버대학교
DAEGU CYBER UNIVERSITY

블록체인 기술과 응용 서비스 4주차

비트코인 블록체인의 구조와 동작원리

1교시 : 비트코인 네트워크 노드와 비트코인 주소 생성

2교시 : 블록의 구조 및 헤더

3교시 : 블록체인 분기

1교시: 비트코인 네트워크 노드와 비트코인 주소 생성

〈학습목표〉

- 비트코인 네트워크 노드(채굴노드, 풀노드, SPV노드)에 대하여 설명할 수 있다.
- 비트코인 주소의 생성 절차에 대하여 설명할 수 있다.

〈주요 용어〉

- **하드코딩**

설정사항이나 코드 등의 시스템적으로 사용하는 변수 값들을 변수에 담아 사용하는 것이 아니라 직접 소스코드에 입력하는 코딩 방식이다.

- **인코딩**

컴퓨터를 이용하여 영상, 이미지 또는 소리 데이터를 생성할 때, 원래의 데이터양을 줄이기 위하여 데이터를 코드화하고 압축하는 것이다.

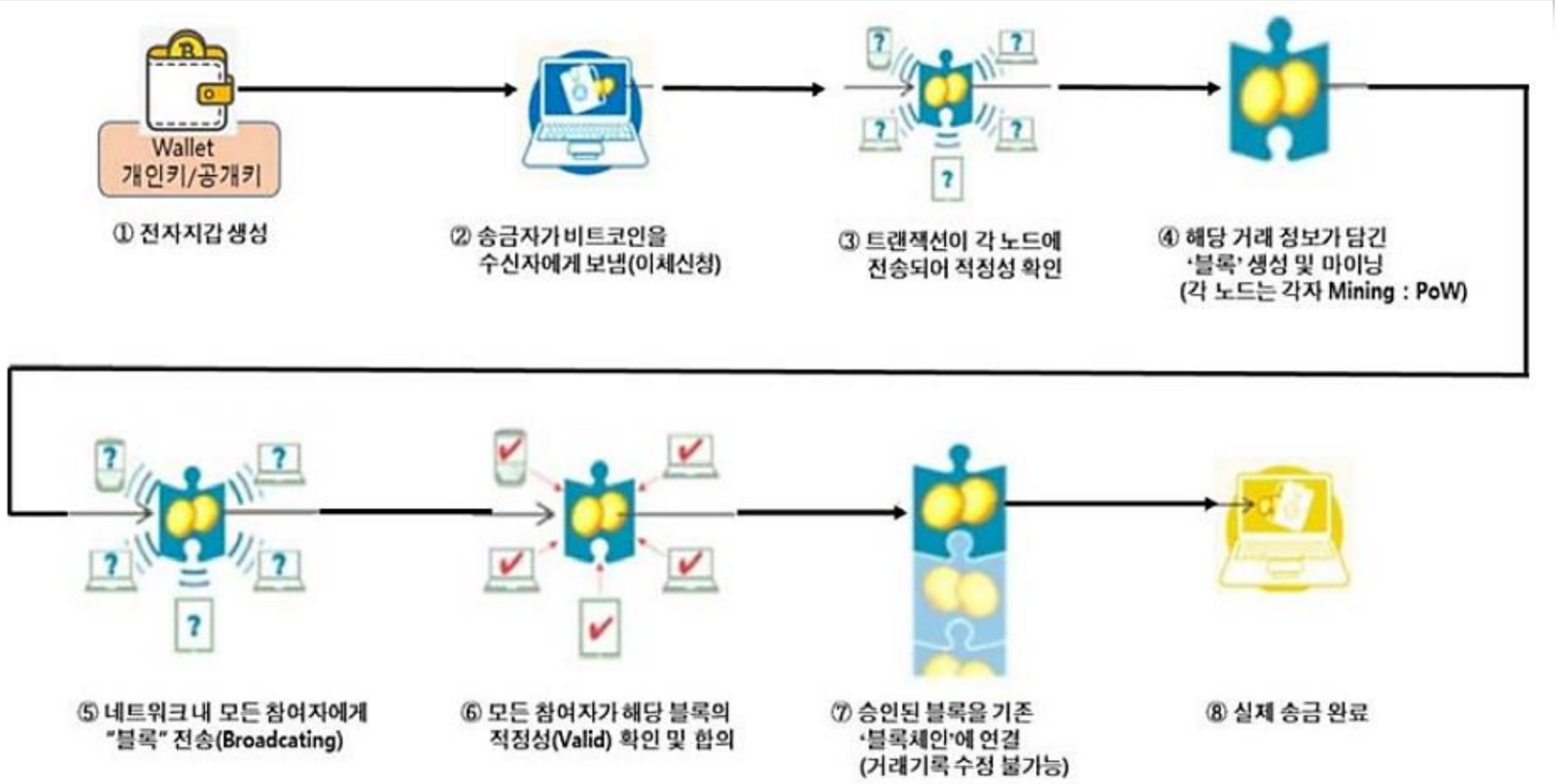
- **베이스58**

큰 숫자를 영숫자(alphanumeric) 텍스트로 나타내기 위해 사용되는 이진 텍스트 인코딩 방식이다.

- **체크섬**

오류 검출 방식의 하나이다. 데이터의 입력이나 전송 시에 제대로 되었는지를 확인하기 위해 입력 데이터나 전송 데이터의 맨 마지막에 모든 데이터를 다 합한 합계를 따로 보내는 것이다. 데이터를 받아들이는 측에서는 하나씩 받아들여 합산한 다음 이를 최종적으로 들어온 검사 합계와 비교하여 착오가 있는지를 점검한다.

비트코인 거래의 흐름도



<자료> Thomson Reuters, 2016. 1.16., 「Blockchain technology: Is 2016 the year of the blockchain?」을 재구성

전자지갑 생성 → 비트코인 전송 → 트랜잭션 전송 및 검증 → 블록생성 → 블록전송 및 검증 → 검증된 블록의 블록체인 연결 → 송금 완료

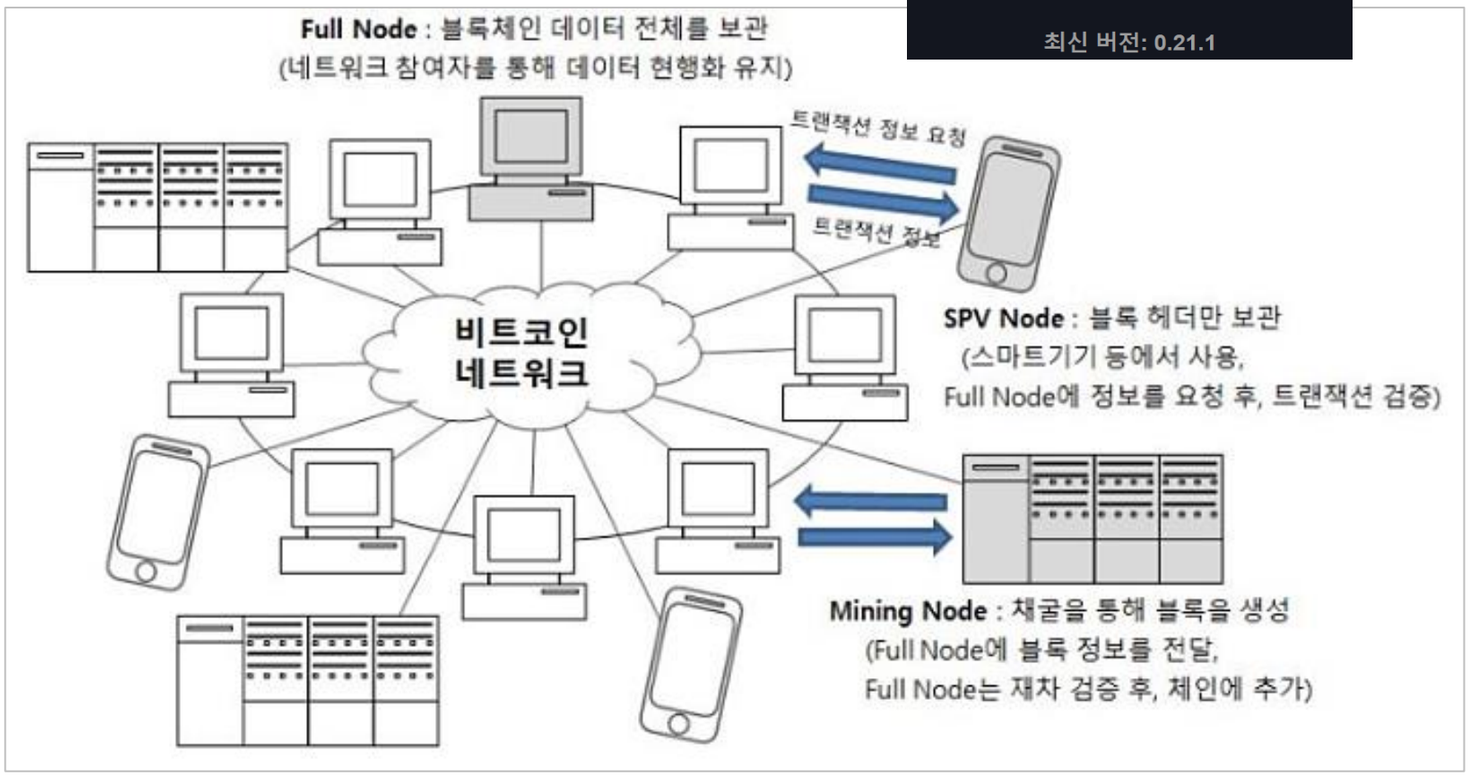
* 출처:
https://www.itfind.or.kr/publication/regular/weeklytrend/weekly/view.do?boardParam1=7457&boardParam2=7457

비트코인 네트워크 노드(1/3)

- **풀노드: 전체 거래내역을 가지며, 상시에 과반 이상의 데이터로 갱신하여 현행화 유지**
  - **채굴을 통해 생성된 블록의 검증 수행**
  - **검증을 담당하는 모든 풀 노드가 채굴노드는 아님**
  - **비트코인 코어(Bitcoin Core): 사토시 나카모토에 의해 배급된 레퍼런스 풀 클라이언트 S/W** (https://bitcoin.org/ko/download)

Bitcoin Core 다운로드

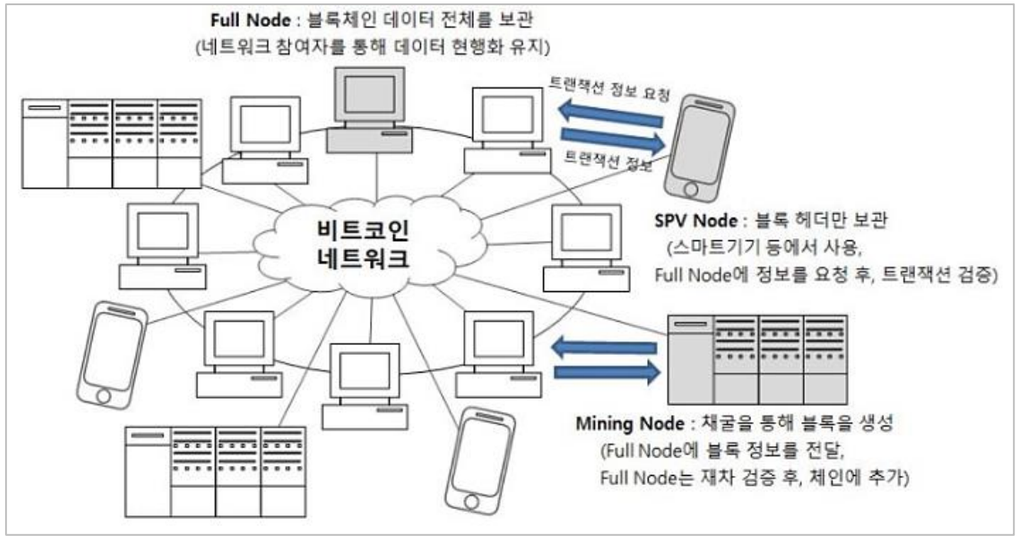
최신 버전: 0.21.1



* 출처 :
https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=jvioonpe&logNo=221572514342

비트코인 네트워크 노드(3/3)

- **블록체인 노드 구축**: 클라이언트 S/W를 실행하여 네트워크에 접속
  - 블록 D/B를 **종자노드**로 부터 다운로드 받아 동기화 후 새로운 노드로 동작
  - **종자노드**(seed node): 네트워크에서 오랜 기간 **안정적으로 작동되고 있는 풀 노드**
  - 추가된 노드는 자신의 IP주소를 이웃 노드들에게 전송하여 다른 노드들이 검색할 수 있게 됨



* 출처 :
https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=jvioonpe&logNo=221572514342

Peer 노드 검색

① 비트코인 클라이언트는 이전에 연결했던 Peer 주소를 캐시에서 가져옴

- Bitcoin Core는 처음 11초 동안 캐시에 있는 Peer들과 연결 시도
- BitcoinJ는 캐시를 사용 않으며, 바로 DNS Seed 방법 사용

② 캐시 방법이 실패하면, 60초 동안 DNS Seed 방법 사용

- 비트코인 커뮤니티가 관리하는 DNS Seed 노드에게 종자노드들의 IP주소 조회
- DNS Seed 노드: Bitseed.xf2.org, Dnsseed.bluematt.me, seed.bitcoin.sipa.be

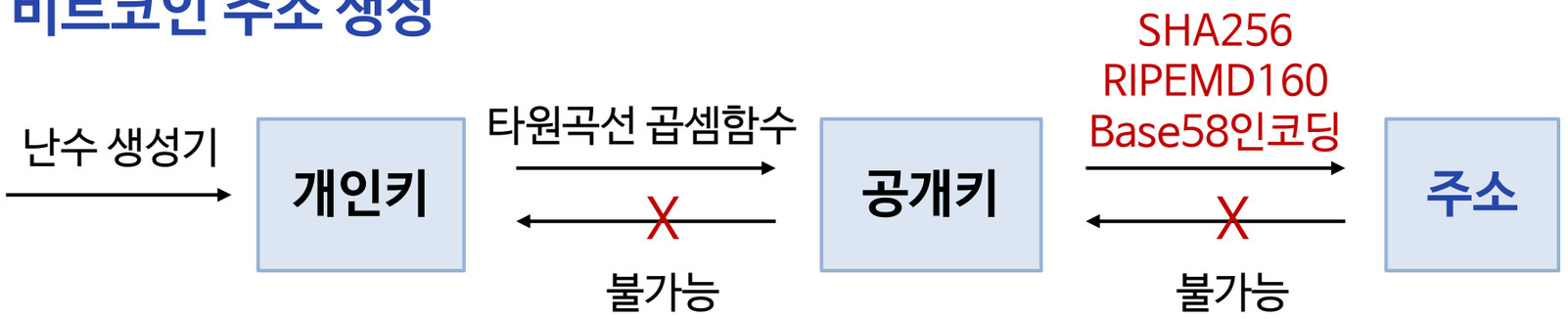
③ 위의 방법들이 실패하면, 마지막으로 chainparamsseeds.h 파일에 하드코드된 IP주소 사용

* 출처 https://www.csharpstudy.com/bitcoin/article/7-%EB%B9%84%ED%8A%B8%EC%BD%94%EC%9D%B8-Peer-%EB%85%B8%EB%93%9C-%EA%B2%80%EC%83%89

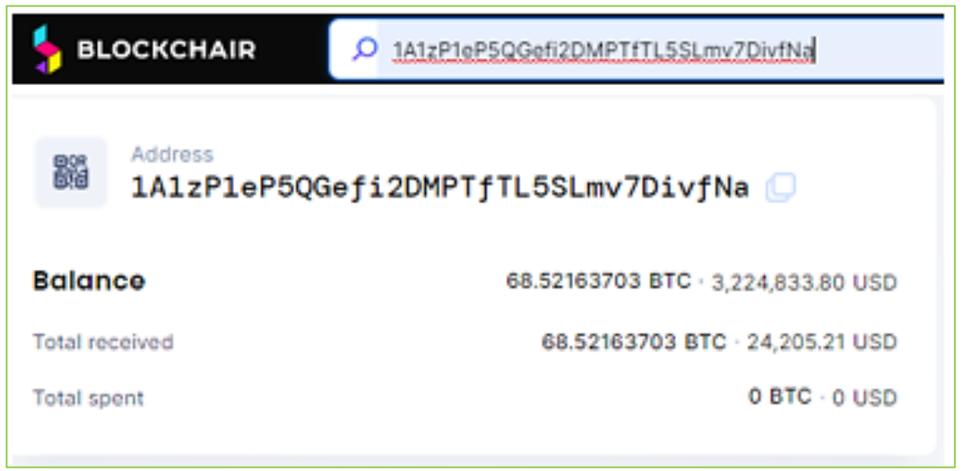
비트코인 주소 생성(1/2)

- 비트코인 계좌를 만들면 자동 생성되는 **개인키**는 송금 시에 본인을 증명하는 **전자서명**에 사용되며, **공개키**는 전자서명의 **검증**에 사용

- **공개키에** 해시연산과 베이스58 인코딩하여 29~35바이트의 **비트코인 주소 생성**



- **사토시의 비트코인 지갑 주소**
  (https://blockchair.com/explorers?from=bitcoin.com)



비트코인 주소 생성(2/2)

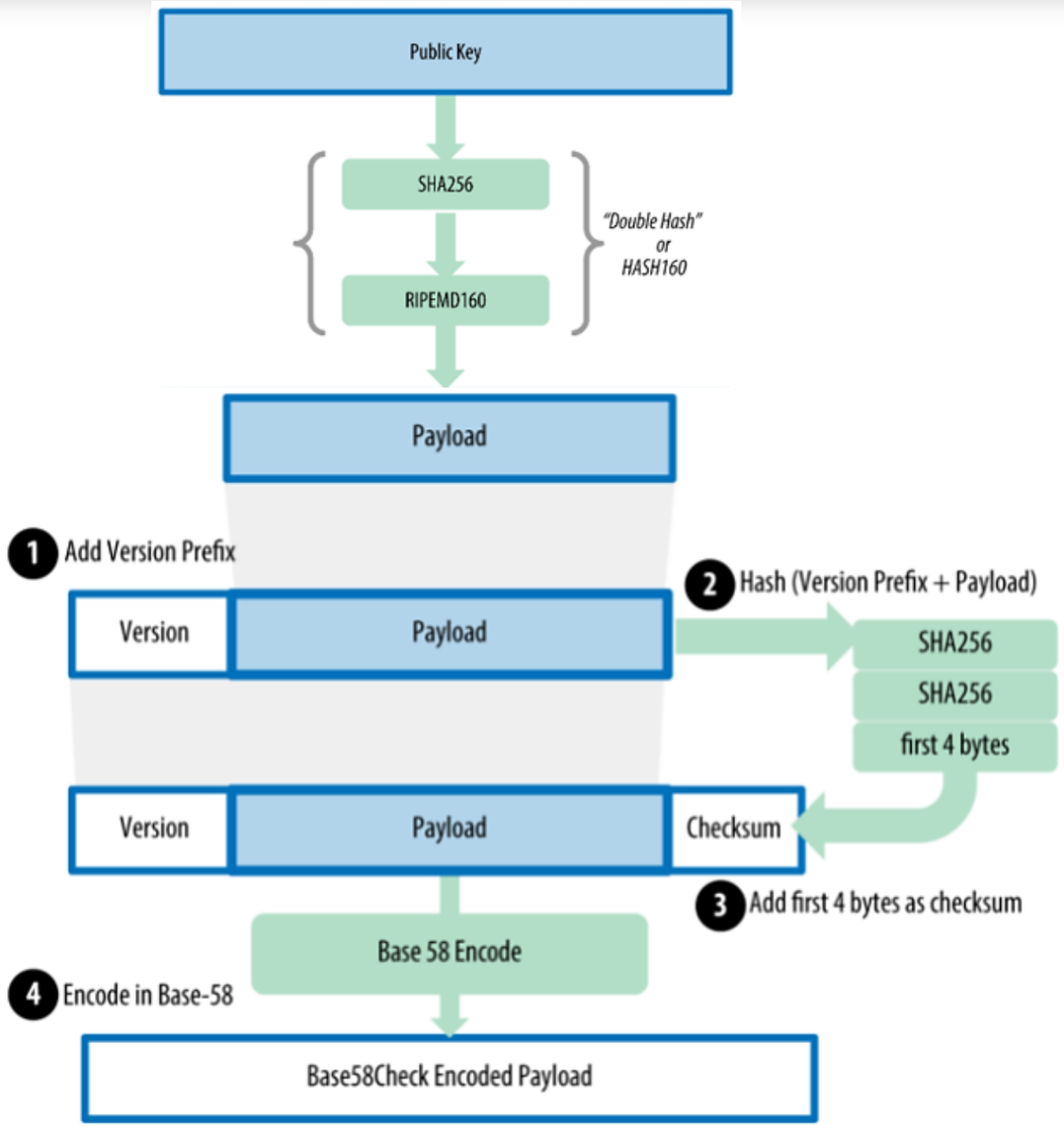
- 공개키 → SHA256 연산 → RIPEMD160 연산 → 160비트 페이로드 생성

① 페이로드에 1바이트 크기의 버전(0x00) 붙임

② SHA256 해시 두 번 수행: SHA256(SHA256(버전 + 페이로드))

③ 2번 결과로 생성된 256비트 중에 첫 32비트를 오류체크 용의 체크섬으로 사용

④ 베이스58인코딩하여 주소 생성
(앞에 있는 0x00 바이트 수만큼 1을 생성하고 나머지 바이트들에 대하여 인코딩)



* 출처 : https://potensj.tistory.com/29

Base58 인코딩

- Base64 인코딩은 사람에게 혼동되는 유사한 문자들이 존재: 숫자 0, 영문 대문자 O, 영문 대문자 I, 영문 소문자 l
- Base58은 Base64에서 혼동되는 숫자와 문자를 제거하고 인코딩

1) 7389 / 58 → 몫 127, 나머지 23(Q)
2) 127 / 58 → 몫 2, 나머지 11(C)
3) 2 / 58 → 나머지 2(3)
4) 7389 → 3CQ

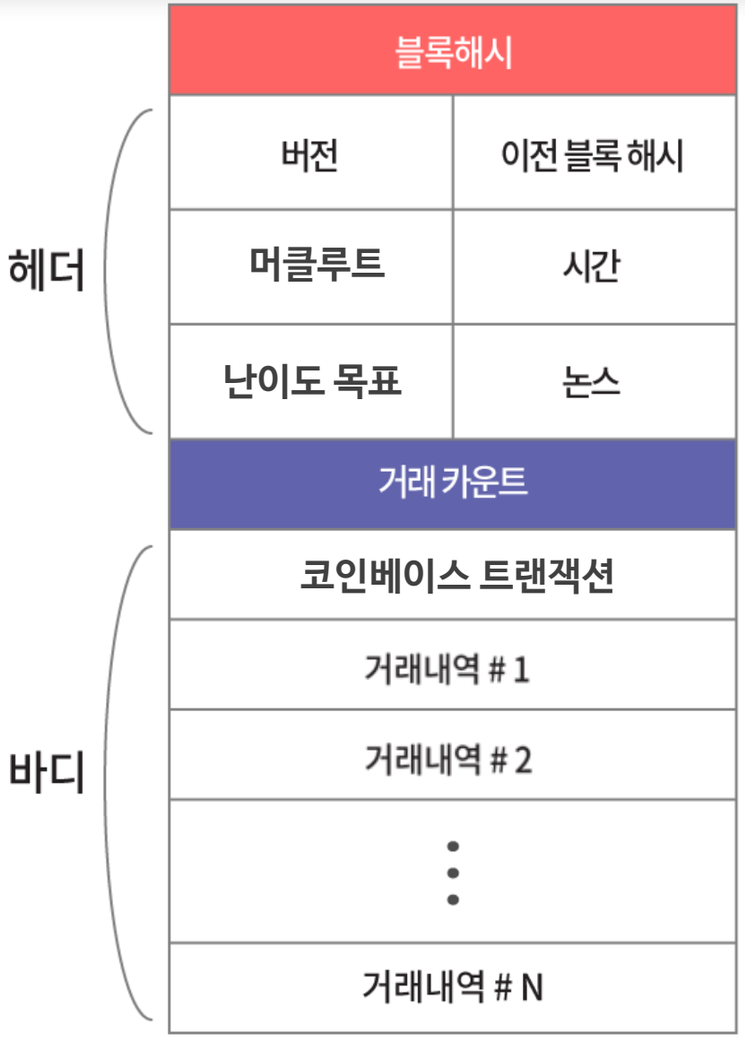
Base58 색인표

Value	Character	Value	Character	Value	Character	Value	Character
0	1	1	2	2	3	3	4
4	5	5	6	6	7	7	8
8	9	9	A	10	B	11	C
12	D	13	E	14	F	15	G
16	H	17	J	18	K	19	L
20	M	21	N	22	P	23	Q
24	R	25	S	26	T	27	U
28	V	29	W	30	X	31	Y
32	Z	33	a	34	b	35	c
36	d	37	e	38	f	39	g
40	h	41	i	42	j	43	k
44	m	45	n	46	o	47	p
48	q	49	r	50	s	51	t
52	u	53	v	54	w	55	x
56	y	57	z				

* 출처 : https://qvault.io/cryptography/base64-vs-base58-encoding/

블록 구조

- **블록해시** (Hash of Block)
  - 블록헤더의 구성요소를 모두 더한 후, SHA-256 함수를 두 번 적용
- **블록헤더** (Header) : 80바이트
- **거래 카운트** (Transaction count) : 블록 해시계산에 사용되지 않음
- **바디** (Body) : 트랜잭션 (거래내역)
  - 평균 1,858개의 거래정보가 포함되며, 평균 1.297Mbyte 크기 (2021.6.30 기준)



* 출처 : https://steemit.com/kr/@yahweh87/3

〈1교시〉 학습정리

- 비트코인 네트워크 노드는 채굴노드, 풀노드, SPV노드로 구분된다. 풀노드는 전체 거래내역을 가지며, 채굴을 통해 생성된 블록의 검증을 수행한다. 검증을 담당하는 모든 풀노드가 채굴노드는 아니다.
- Peer 노드 검색을 위하여 이전에 연결했던 Peer 주소를 캐시에서 가져오며, 캐시 방법이 실패하면, 60초 동안 DNS Seed 방법을 사용한다. 캐시와 Seed 방법들이 실패하면, 마지막으로 chainparamsseeds.h 파일에 하드코드된 IP주소를 사용한다.
- 비트코인 계좌를 만들면 자동 생성되는 개인키는 송금 시에 본인을 증명하는 전자서명에 사용되며, 공개키에 해시연산과 베이스58 인코딩하여 29~35바이트의 비트코인 주소를 생성한다.

〈1교시〉 학습평가

1. 비트코인 네트워크 노드의 설명 중 틀린 것을 고르시오.

1) 비트코인 코어는 사토시 나카모토에 의해 배급된 풀노드로 레퍼런스 클라이언트이다.

2) 풀노드는 전체 거래내역을 가지며, 채굴을 통해 생성된 블록의 검증을 수행한다. 검증을 담당하는 모든 풀노드가 채굴노이다.

3) SPV노드는 블록헤더만 보관하며, 거래의 검증은 풀노드에 정보를 요청 후 트랜잭션을 검증한다.

4) 블록체인의 종자노드는 네트워크에서 오랜 기간 안정적으로 작동되고 있는 풀노드이다.

답) 2

해설) 풀노드는 전체 거래내역을 가지며, 채굴을 통해 생성된 블록의 검증을 수행한다. 검증을 담당하는 모든 풀노드가 채굴노드는 아니다.

2. 비트코인 주소 생성 절차에 대하여 설명하시오.

해설) - 난수 생성기를 이용하여 개인키를 생성 후, 타원곡선 곱셈함수로 공개키를 생성한다. 생성된 공개키에 해시연산과 베이스58 인코딩하여 29~35바이트의 비트코인 주소를 생성한다.

2교시: 블록의 구조 및 헤더

〈학습목표〉

- 비트코인 블록체인의 블록 구조(헤더와 바디)에 대하여 설명할 수 있다.
- 비트코인 블록의 트랜잭션과 채굴 난이도에 대하여 설명할 수 있다.

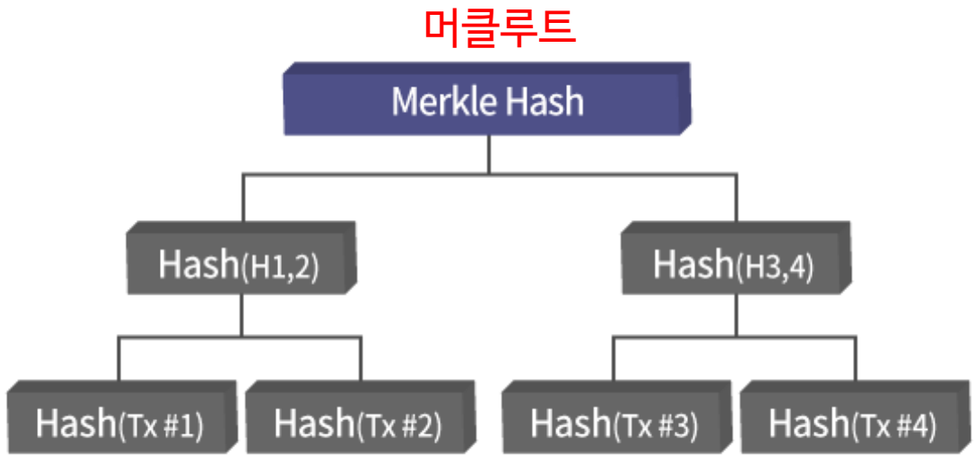
블록 헤더

- 버전(Version)
- 이전 블록해시(Previous block hash)
- 머클루트(Merkle hash root)
- 시간(Timestamp): 블록 생성 시간으로 1970년 1월 1일 이후의 초 단위 시간
- 난이도목표(Bits): 작업증명의 해시 난이도 목표 값
- 논스(Nonce): 난이도목표 값 이하를 구할 때 입력 값으로 사용

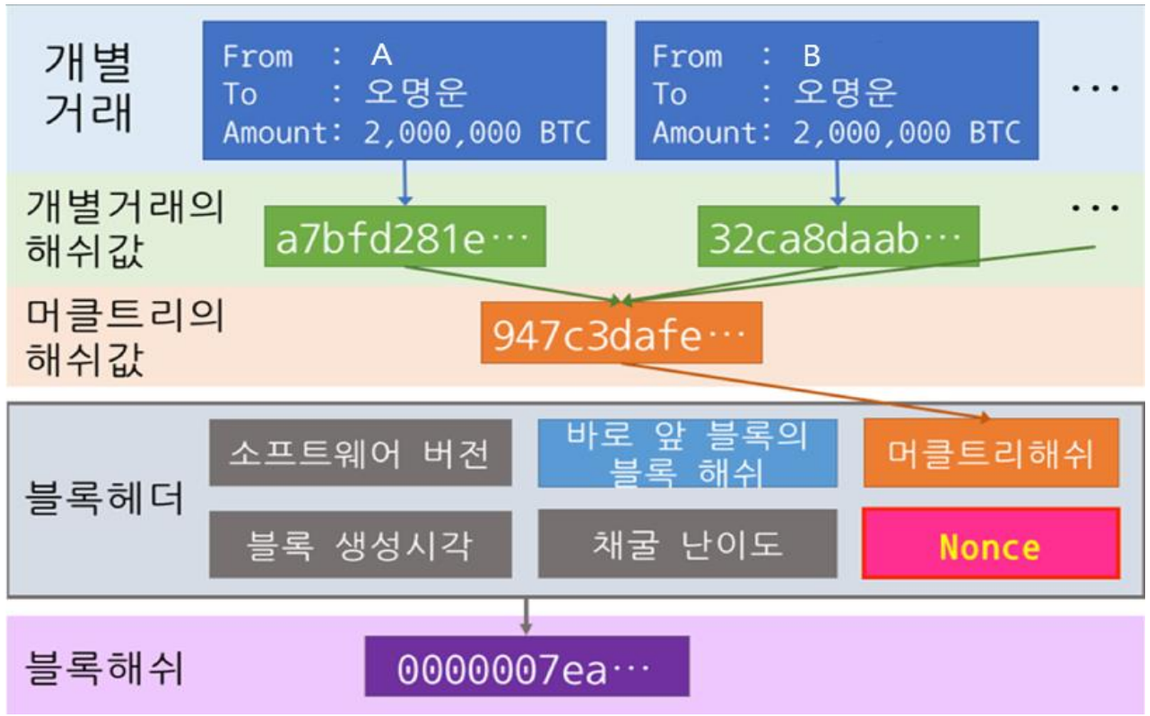
머클해시루트

- 트랜잭션 i 해시값과 트랜잭션 i+1 해시값을 더해서 **해시값, H(i, i+1)** 계산
  - 각각의 트랜잭션을 **이진트리** 형태로 만들 경우 최종적으로 남는 **루트 해시값**

- 거래 데이타가 하나라도 바뀌면 **머클루트가 변함** → 머클루트 값을 비교하여 거래가 위변조 되었는지 **효율적 검증**이 가능



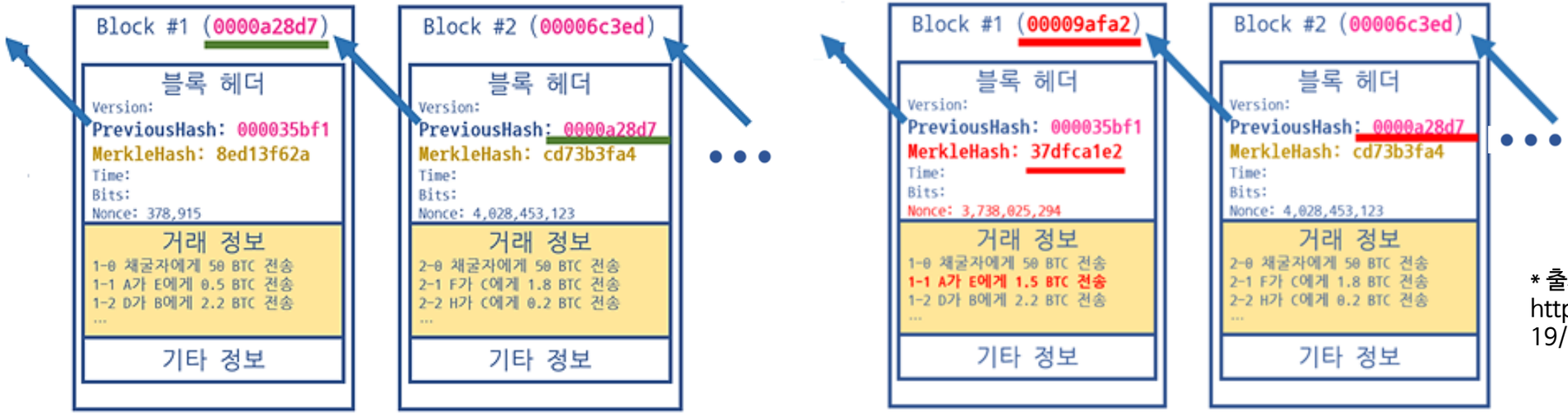
출처:
https://m.blog.naver.com/PostView.nhn?blogId=renucs&logNo=220958282185&proxyReferer=https%3A%2F%2Fwww.google.com%2F



출처:
https://homoefficio.github.io/2017/11/19/%EB%B8%94%EB%A1%9D%EC%B2%B4%EC%9D%B8-%ED%95%9C-%EB%B2%88%EC%97%90-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0/

이전 블록 해시

- 블록체인은 이전 블록해시 값으로 블록 간의 연결이 이루어짐
- 블록 1의 어떤 **거래가 변경**되면 머클루트 해시 값 변경 → **블록 1의 블록해시가 변경되어** 블록 2의 이전블록해시 값과 일치하지 않음
  - 체인을 유지하려면 **블록 2의 이전블록해시 값 갱신 후에 블록해시를 새로 계산**하고, 연이어 이후의 모든 블록을 순서대로 다시 채굴해야 함
  - 거래를 변조하는 10분 동안 다른 노드들은 새로운 블록 생성 → 길이가 긴 정상 블록체인에 의해 **짧은 악의적인 블록체인은 버려짐**

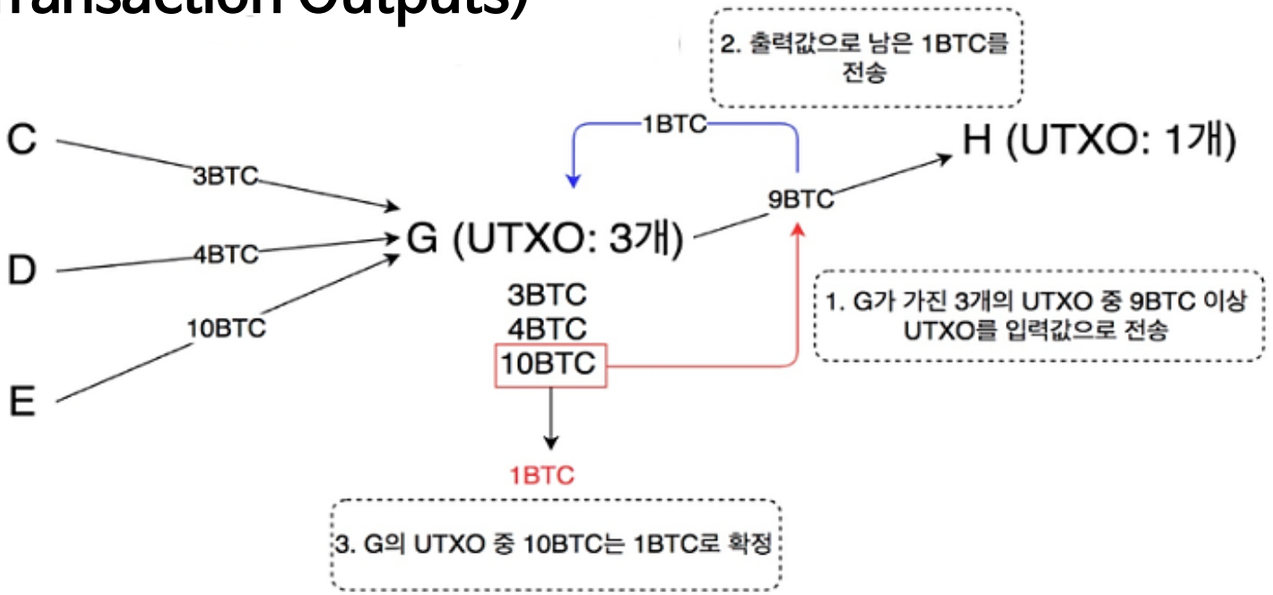


* 출처 : https://homoefficio.github.io/2017/11/19/블록체인-한-번에-이해하기/

UXTO(1/2)

- **미사용 트랜잭션 출력 값**(UTXO: Unspent Transaction Outputs)

- 지갑주소 G에서 H로 9BTC 송금
  - ① G의 UTXO 중 9BTC 이상인 UTXO 10BTC 선택 → H로 9BTC 송금
  - ② 남은 1BTC를 G 지갑으로 전송
  - ③ G의 UTXO(10BTC)는 1BTC로 확정

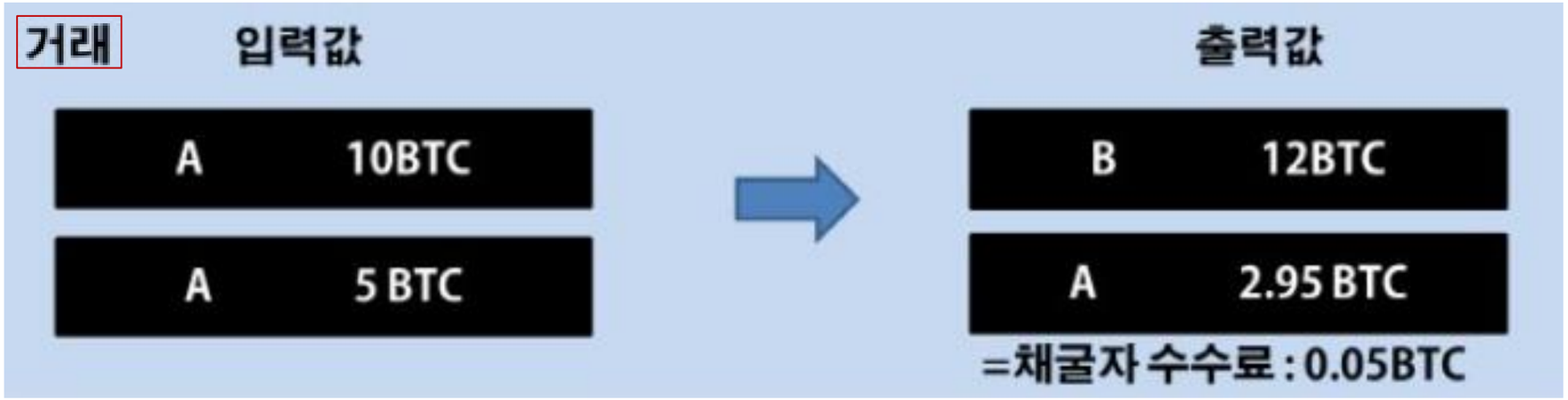


* 출처 : https://m.blog.naver.com/mage7th/221439829511

- 비트코인은 이더리움의 계좌잔고모델(Account Balance Model)과 달리 **계정이나 잔고가 없음**
  - 블록체인 기록 중에 내 앞으로 보내진 UTXO를 찾아 모으면 나의 잔고가 됨
  - 지갑업체의 기본 서비스는 이용자들이 편리하게 송금하고 잔고를 확인하는 UI를 제공함

UTXO(2/2)

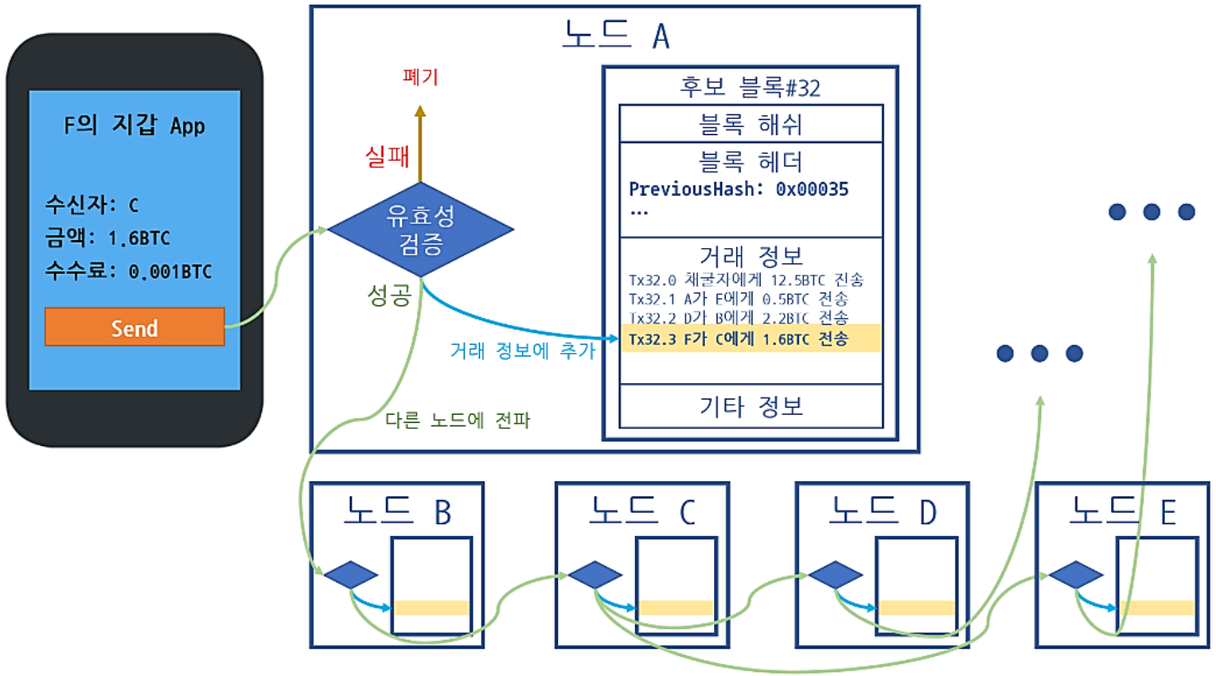
- 거래에서 소비하고자 하는 UTXO가 **입력값**이며, 새로 만들어지는 UTXO가 **출력값**
- 지갑주소 A에서 10BTC와 5BTC 두 개의 UTXO를 사용하여 B로 12BTC 송금
  - 지갑 A에서 10BTC와 5BTC 소비되었으므로 **더 이상 UTXO가 아님**
  - A에게 잔금을 송금하지 않으면 **3BTC가 모드 수수료**로 지불됨
  - **2.95BTC를 다시 A로 송금 → 수수료 0.05BTC**가 수수료로 자동 지불됨
  - **두개의 UTXO가 새로 탄생**: B앞으로 된 12BTC와 A로 보낸 2.95BTC



출처 : https://m.blog.naver.com/mage7th/221439829511

트랜잭션 정보의 전파

- 사용자 F가 C에게 1.6BTC와 수수료 0.001BTC 송금 → 지갑 앱은 블록체인 네트워크 상의 노드 A에게 거래정보 전송
  - ① 노드 A는 **해당 거래의 유효성 검증** 후에 그 거래를 생성 대기 중인 **후보 블록에 추가**하고, 인접한 노드로 전파
  - ② 전파 받은 노드 B도 A와 같은 일을 수행 후 다른 노드에게 전파하며, 이 거래는 결국 전체 노드로 전파



* 출처 :
https://i.imgur.com/QPLJbWM.png

트랜잭션의 검증

- 각 노드는 독립적으로 검증 리스트에 따라 수신한 트랜잭션을 검증 후, 이웃 노드로 브로드캐스팅하며 올바르지 않은 트랜잭션은 제거

트랜잭션 검증 리스트
① 트랜잭션 구문과 데이터 구조 확인
② 트랜잭션 입력 값은 반드시 UTXO인 것을 확인
③ 트랜잭션 크기가 100 바이트보다 크고 1.297Mbyte 작은 지 확인
④ 트랜잭션 입력 값이 출력 값보다 작은지 확인

트랜잭션 구조

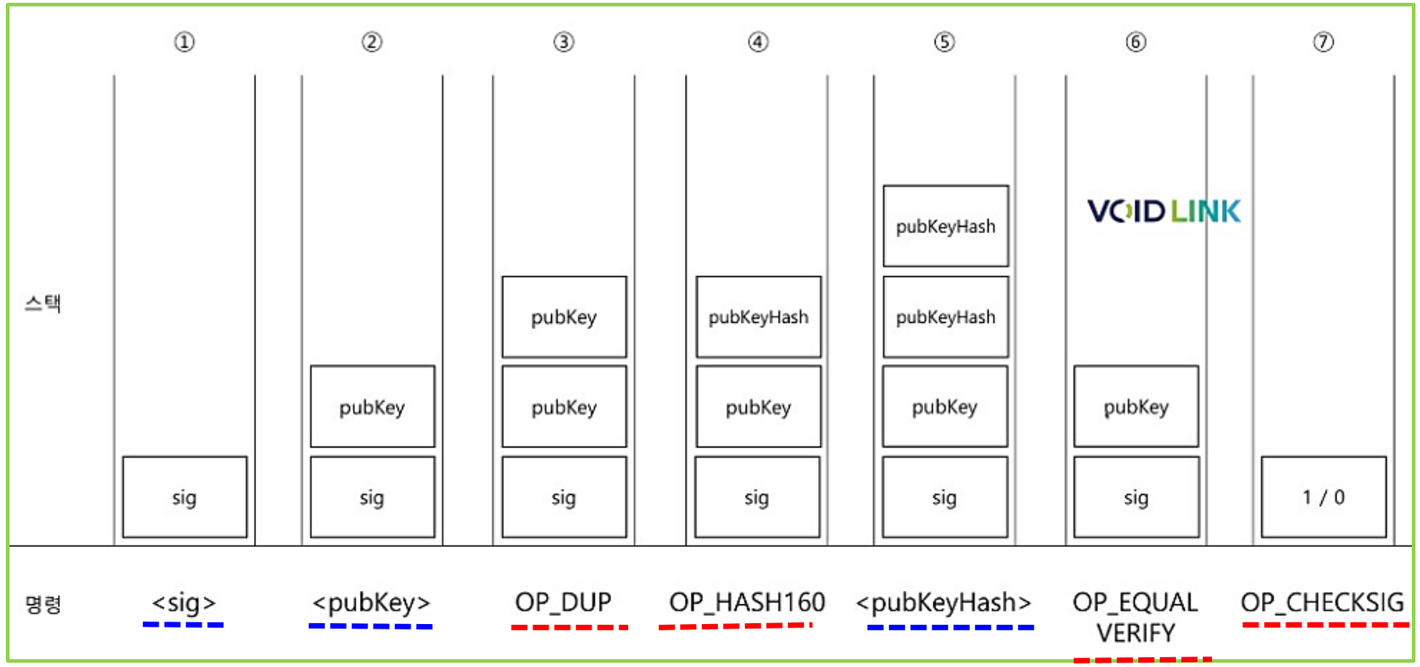
- 블록의 바디에는 첫 번째 나타나는 코인베이스 트랜잭션과 하나 이상의 일반 트랜잭션으로 구성
  - 코인베이스 트랜잭션에는 블록의 채굴자에게 지급하는 보상금 내역이 있음

트랜잭션	
Txid: 32바이트 해시값, Version, Size	
Locktime: 트랜잭션 잠금 기간으로 0이면 바로 네트워크로 전파	
입력부분	출력부분
Input Counter	Output Counter
- Txid: 지출하려는 코인의 출처	- Value: 송금하는 금액
- ScriptSig: 전자서명 + 공개키	- Address: 수신 비트코인 주소
	- ScriptPubKey: 서명 검증정보 및 방법
. . .	. . .

ScriptPubKey: 서명(sig), 공개키(pubKey),
OP_DUP, OP_HASH256, 공개키해시(pubKeyHash), OP_EQUALVERIFY, OP_CHECKSIG

트랜잭션의 전자서명 확인

① 송금자의 서명(sig)와 ② 송금자의 공개키(pubKey)를 스택에 넣음
③ OP_DUP: pubKey를 복사하여 스택에 넣음 ④ OP_HASH160: pubKey의 해시를 계산하여 pubKeyHash로 스택에 넣음 ⑤ 송금자의 공개키해시(pubKeyHash)를 스택에 넣음
⑥ OP_EQUALVERIFY: pubKeyHash 2 개를 비교하여 일치하지 않으면 강제종료 ⑦ OP_CHECKSIG: 송금자의 pubKey로 sig를 검증하여 결과(1 또는 0)을 스택에 넣음



스크립트 <sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG

* 출처 : https://medium.com/@jaewoo/5-4-비트코인의-utxo와-잔액-f4805502fae8

〈2교시〉 학습정리

- 비트코인 블록체인의 블록에는 블록해시, 블록헤더와 바디가 포함된다.
- 80바이트의 블록헤더에는 6개의 필드(버전, 이전 블록해시, 머클루트, 시간, 난이도, 논스)가 포함되며, 바디에는 평균 1858개의 거래정보가 포함(2021.6.30. 기준)된다.
- 비트코인은 이더리움의 계좌잔고모델과 달리 계정이나 잔고가 없으며, 블록체인 기록 중에 내 앞으로 보내진 미사용 트랜잭션 출력 값(UTX)을 찾아 모으면 나의 잔고가 된다.
- 난이도 목표는 새로운 블록을 생성하기 위해 계산하는 해시값의 어려움 정도이다. 2,016개 블록의 생성 주기로 블록의 평균 생성 시간이 10분 보다 오래 걸리면 낮추고 적게 걸리면 난이도를 높인다.

〈2교시〉 학습평가(1/2)

1. 비트코인 블록체인의 구조에 대한 설명 중 틀린 것을 고르시오.

1) 2,016개 블록의 생성 주기로 블록의 평균 생성 시간이 10분 보다 오래 걸리면 난이도를 낮추고 적게 걸리면 난이도를 높인다.

2) 블록체인은 링크드 리스트로 이전 블록해시 값으로 블록 간의 연결이 이루어진다.

3) 블록해시는 거래 카운트와 블록헤더의 구성요소를 모두 더한 후, SHA-256 함수를 두 번 적용한다.

4) 거래 데이타가 하나라도 바뀌면 머클루트가 변하므로 머클루트 값을 비교하여 거래가 위변조 되었는지 효율적 검증이 가능하다.

답) 3

해설) 블록해시는 블록헤더의 구성요소를 모두 더한 후, SHA-256 함수를 두 번 적용한다. 단, 거래 카운트는 블록 해시계산에 사용되지 않는다.

2. 트랜잭션의 검증 리스트에 대하여 설명하시오.

해설) - 트랜잭션의 검증 리스트

- o 트랜잭션 구문과 데이터 구조 확인
- o 트랜잭션 입력 값은 반드시 UTXO인 것을 확인
- o 트랜잭션 크기가 100 바이트보다 크고 1.297Mbyte 작은 지 확인
- o 트랜잭션 입력 값이 출력 값보다 작은지 확인

〈2교시〉 학습평가(2/2)

3. 트랜잭션의 입력 및 출력 부분에 대하여 설명하시오.

해설) – 트랜잭션의 입력 및 출력 부분

o 입력부분에는 Input Counter, Txid(지출하려는 코인의 출처), ScriptSig(전자서명 + 공개키)로 구성된다.

o 출력부분에는 Output Counter, Value(송금하는 금액), ScriptPubKey(서명 검증정보), Address(수신 비트코인 주소)로 구성된다.

3교시: 블록체인 분기

〈학습목표〉

- 블록체인의 분기가 발생하는 이유 및 해결되는 과정에 대하여 설명할 수 있다.
- 블록체인의 시연을 통하여 키쌍 생성, 트랜잭션 전자서명, 블록해시, 블록체인의 생성에 대하여 설명할 수 있다.

채굴

❑ **작업증명**(PoW: Proof of Work)을 통해 블록을 생성한 **노드(채굴자)에게 코인과 수수료**를 보상으로 지급

- 21만개의 블록이 생성되는 약 4년 주기로 보상금이 절반으로 감소:
  4기(2020~)는 6.25, 5기(2024 ~)는 3.125BTC
- 거래가 블록에 추가되는 순위를 결정하는데 거래 **수수료가 입력 값으로 사용됨**

채굴(Mining)

=

작업증명(Pow)

+

보상(Reward)

출처: www.gettyimagesbank.com/

난이도 목표(1/2)

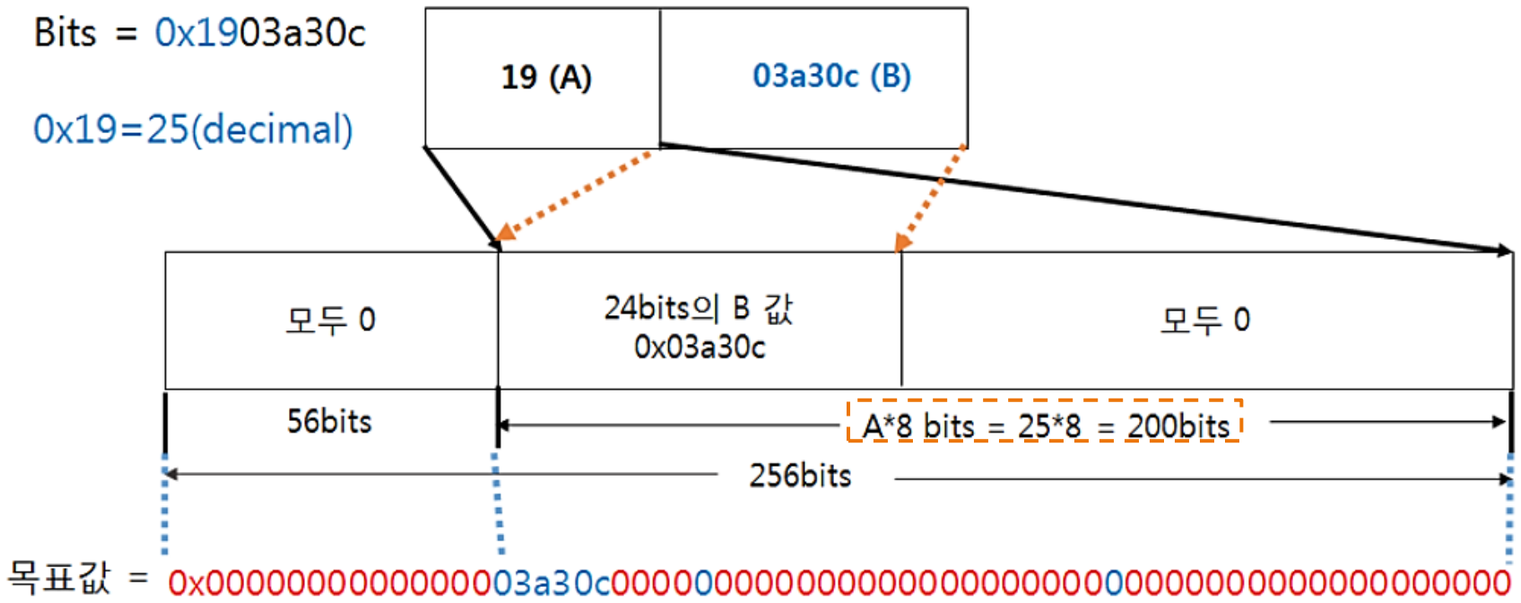
❑ 상자에 있는 포켓볼 15개 중 1개를 뽑았을 때 그 번호가
- 1보다 작거나 같을 확률: 1/15
- 5보다 작거나 같을 확률: 5/15

❑ 난이도 목표: 새로운 블록을 생성하기 위해 계산하는 해시값의 어려움 정도
- 2,016개 블록의 생성 주기로 블록의 평균 생성 시간이 10분 보다 오래 걸리면 낮추고 적게 걸리면 난이도를 높임

출처: https://apkpure.com/%EC%A0%9C%EB%B9%84%EB%BD%91%EA%B8%B0/com.companyname.jabibbobghi
https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=snowjy79&logNo=220193035130&view=img_3

난이도 목표(2/2)

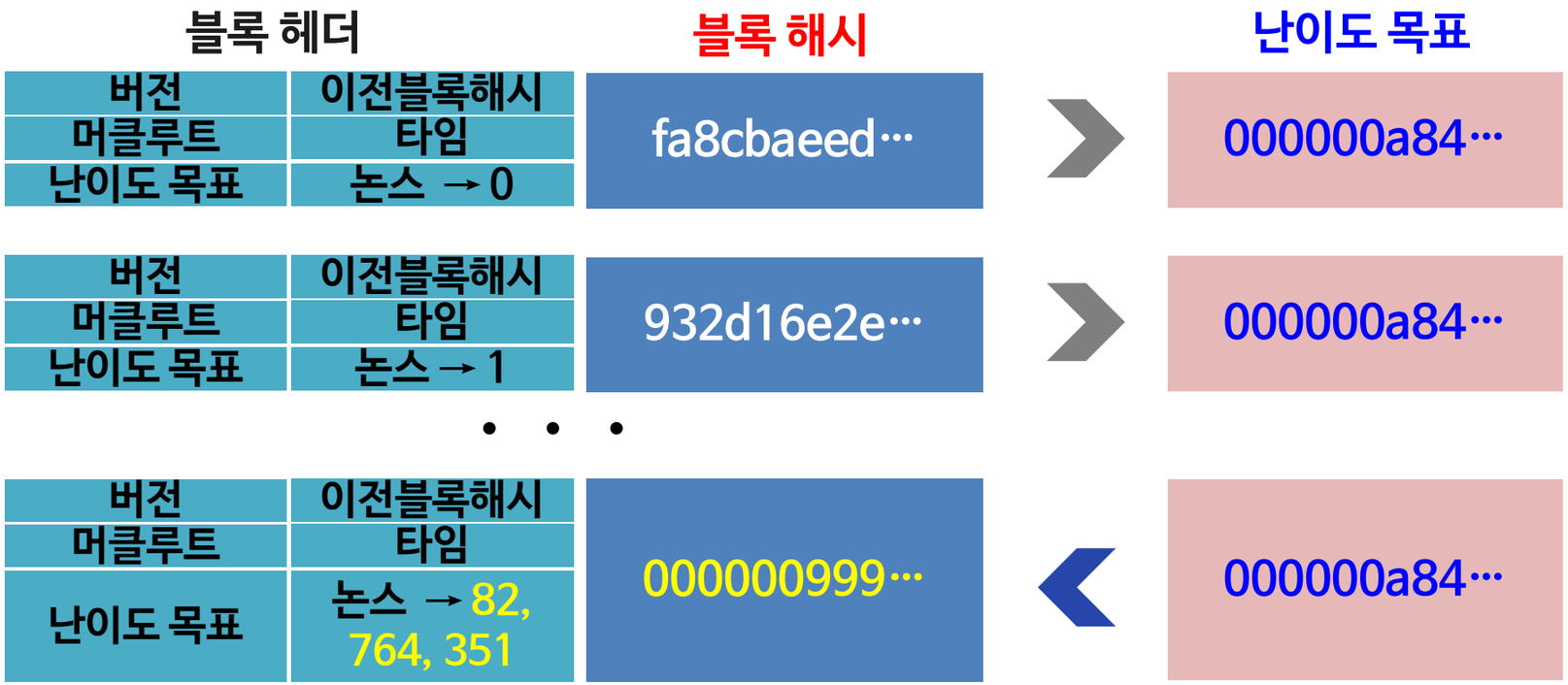
- **난이도 목표(Bits) : 현재의 난이도 목표 값(CURRENT_TARGET)**



* 출처 : 비트코인 블록체인 동작원리 및 진화, 주간기술동향 2018.6.20

임시 값(논스)

- 논스(Nonce): 블록해시의 생성 조건을 맞추기 위해 사용하는 임시 값
  - ① 논스를 적용하여 블록해시 계산
  - ② 블록해시가 난이도 목표값 이하면 작업증명(채굴) 완료
  - ③ 난이도 목표값 이상이면 논스를 1 증가 후, 1번 작업으로 분기



블록체인의 분기(1/4)

❑ 블록의 생성 및 전파

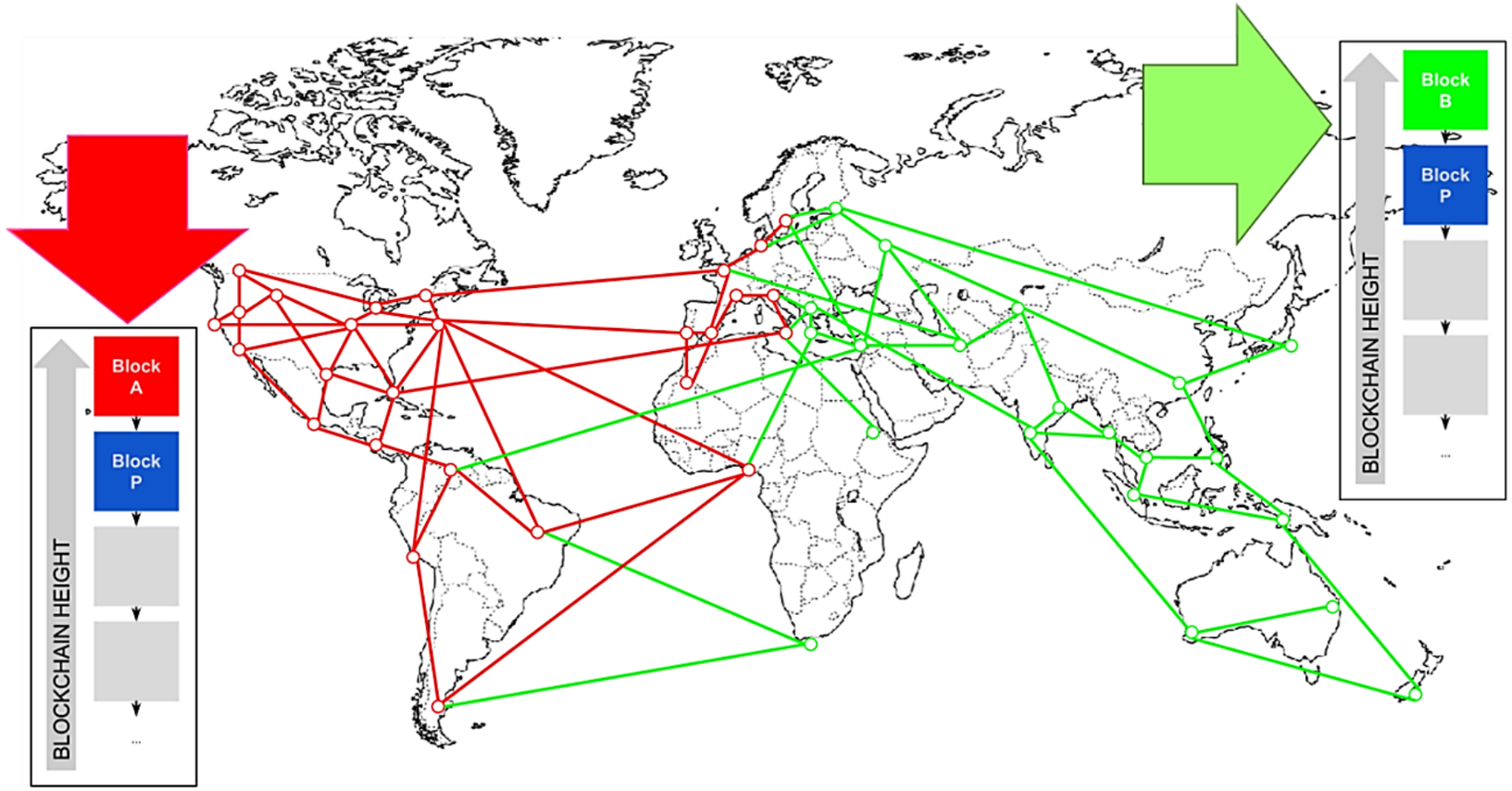
- 거래 정보가 채워지면 각 노드는 블록을 생성
- 캐나다 노드와 호주 노드는 멀리 떨어져 있으므로, 각 블록에 담겨 있는 거래의 내용과 순서는 서로 다를 수 있음

출처 : https://i.imgur.com/AraOlPF.png

블록체인의 분기(2/4)

- 마지막 파랑 블록 다음에 캐나다와 호주 지역의 노드가 **동시에** nonce 값을 찾아서 **새로운 블록을 생성** 후 전파

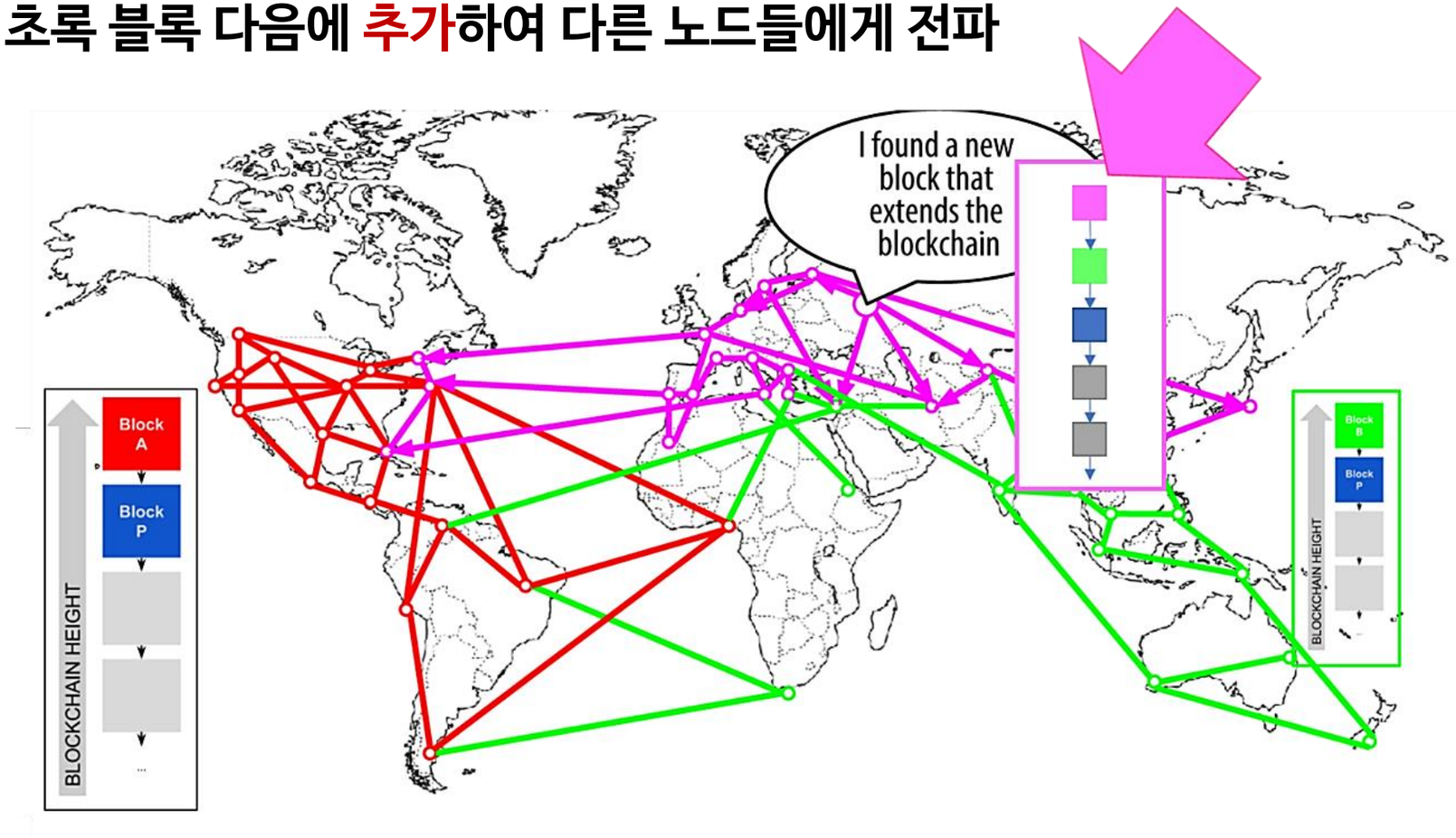
- 인접 노드들은 **검증 후 블록 추가**



* 출처 : https://i.imgur.com/6ceiHDs.png

블록체인의 분기(3/4)

- 포르투갈의 노드에 빨강 블록이 먼저 전파된 후에 초록 블록 전파 → 늦게 도착한 초록 블록 무시
- 러시아 지역의 노드는 빨강 블록이 늦게 도착하여 초록 블록 채택 → 새로 생성한 분홍 블록을 초록 블록 다음에 추가하여 다른 노드들에게 전파

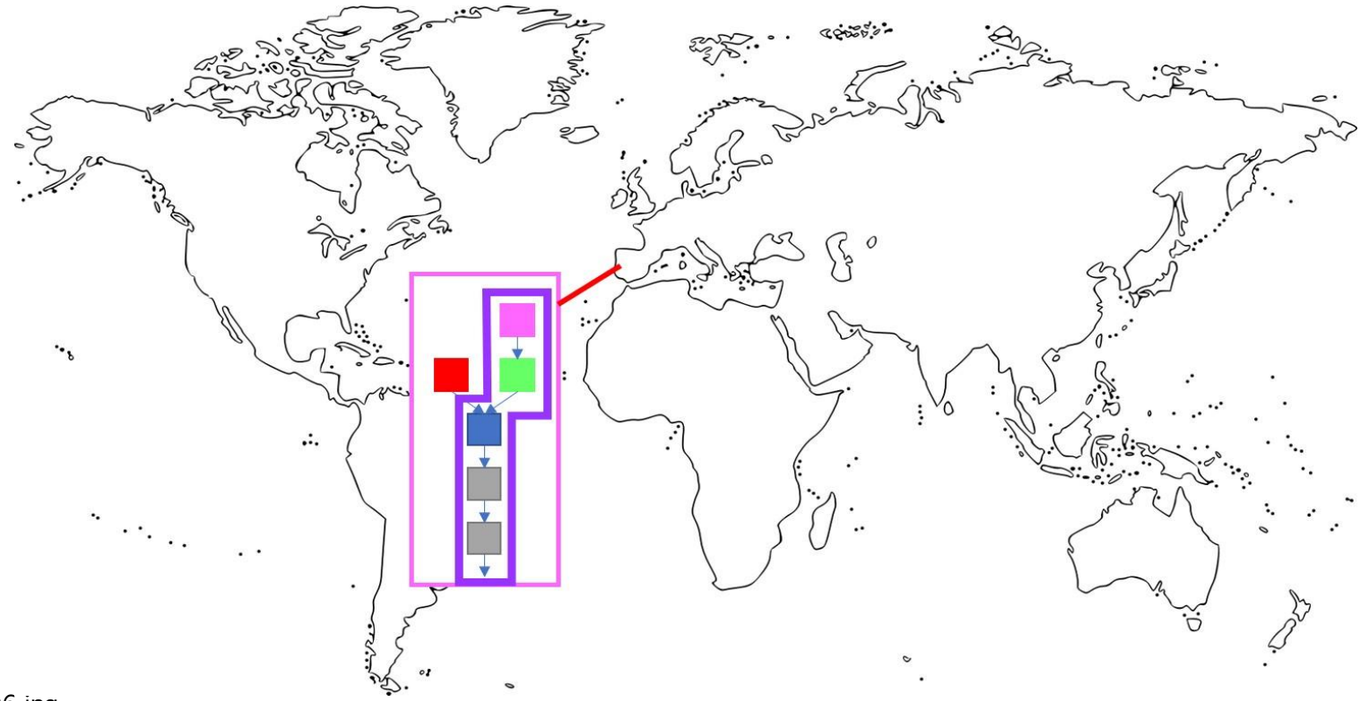


* 출처 : https://i.imgur.com/VCQiHbJ.png

블록체인의 분기(4/4)

- 빨강 블록을 선택한 포르투갈 노드는 파랑, 초록, 분홍으로 이어진 블록을 받아 **블록체인의 분기 발생**
  → 동기화를 위하여 **길이가 긴 체인 선택**

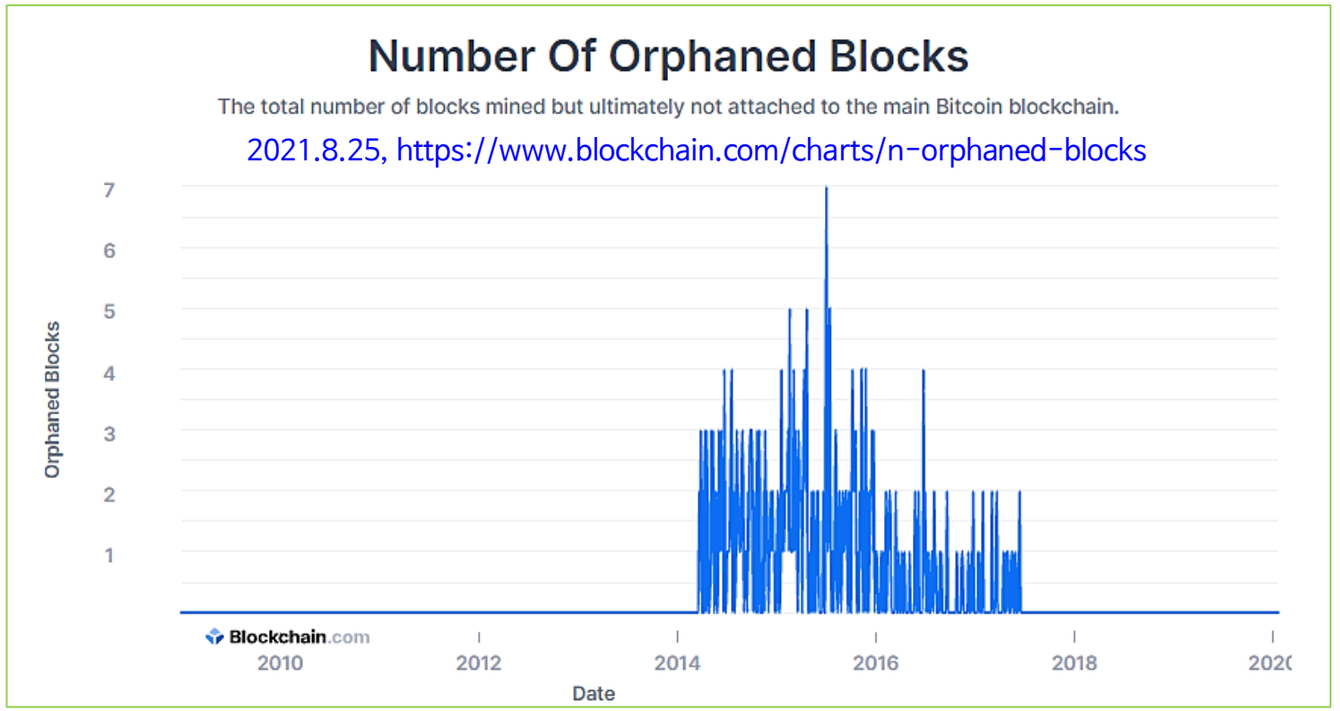
- **고아가 된 빨강 블록안의 거래 T는 소멸되나요?**
  - 블록체인에 포함되지 않은 다른 거래처럼 취급되어 초록 블록에 없더라도 분홍 또는 이후의 블록에 **결국 포함됨**



출처 : https://i.imgur.com/7fF9iu6.jpg

고아 블록의 수

- 블록체인에 의해 고아(Orphand)가 되는 블록의 수는 2017년 이후 발생하지 않음
- 2~7개의 블록이 추가되는 과정에서 일시적으로 발생한 분기 상태가 해소되고 결국 하나의 블록체인만 남음
  - 블록 생성은 평균 10분이 소요되는 연산량이 큰 작업 → 동시에 다른 블록이 생성되어 분기될 가능성은 아주 적음



출처: blockchain.com

〈3교시〉 학습정리

- 거리가 멀리 떨어져 있는 노드에서 각 블록에 담겨 있는 거래의 내용과 순서는 서로 다를 수 있으며, 이로 인해 블록체인의 분기가 발생할 수 있다.

- 비트코인 블록체인에서는 2~7개의 블록이 추가되는 과정에서 일시적으로 발생한 분기 상태가 해소되고 결국 하나의 블록체인만 남는다. 블록 생성은 평균 10분이 소요되는 연산량이 큰 작업이므로 동시에 다른 블록이 생성되어 분기될 가능성은 아주 적다.

〈3교시〉 학습평가

1. 비트코인 블록체인의 분가가 발생하는 이유에 대하여 기술하시오.

해설) - 거리가 멀리 떨어져 있는 노드에서 각 블록에 담겨 있는 거래의 내용과 순서는 서로 다를 수 있으며, 이로 인해 블록체인의 분기가 발생할 수 있다. 블록 생성은 평균 10분이 소요되는 연산량이 큰 작업이므로 동시에 다른 블록이 생성되어 분기될 가능성은 아주 적다.

2. 비트코인 블록체인의 고아 블록에 대하여 설명하시오,

해설) 블록체인의 분기 과정에서 고아 블록이 발생한다. 2~7개의 블록이 추가되는 과정에서 일시적으로 발생한 분기 상태가 해소되고 결국 하나의 블록체인만 남으며 고아 블록은 무시된다.