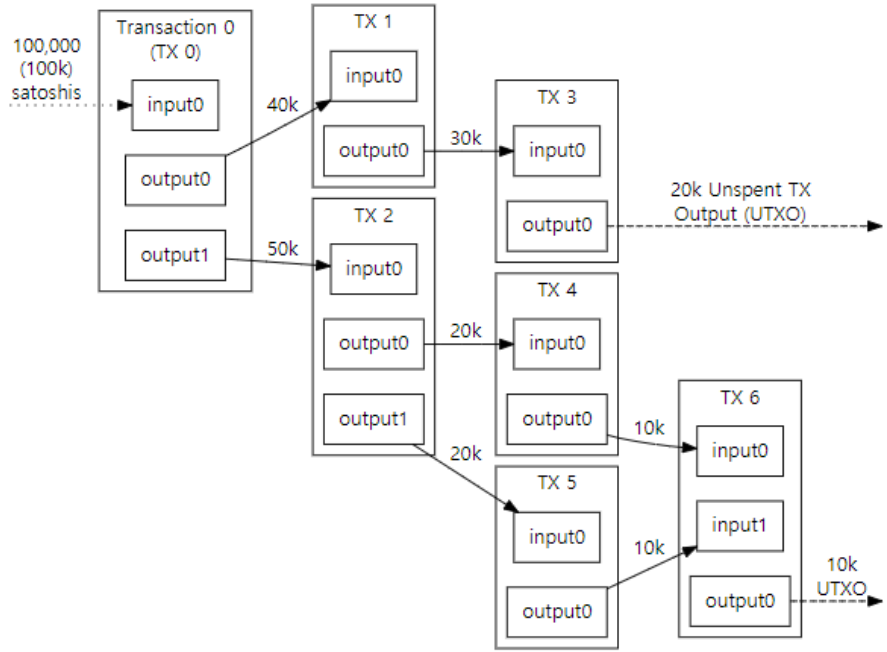
블록체인 #3

서울대학교 산업공학과 장우진

Contents

1) 비트코인 Transaction 과정



그림출처: https://bitcoin.org/en/developer-guide#block-chain-overview

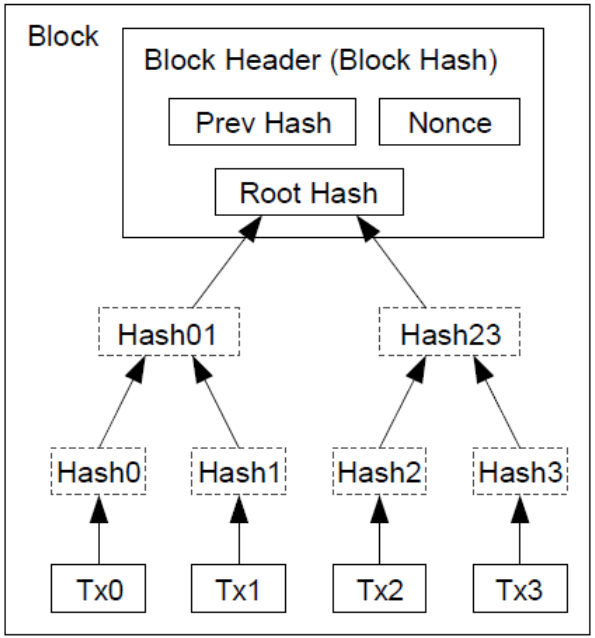
2) 비트코인 Transaction Script


```
{
        "hash":"5a42590fbe0a90ee8e8747244d6c84f0db1a3a24e8f1b95b10c9e050990b8b6b",
        "ver":1,
        "vin_sz":2,
        "vout_sz":1,
        "lock_time":0,
        "size":404,
        "in":[
         {
          "prev_out":{
           "hash":"3be4ac9728a0823cf5e2deb2e86fc0bd2aa503a91d307b42ba76117d79280260",
           "n":0
          },
            "scriptSig":"30440..."
         },
         {
          "prev_out":{
           "hash":"7508e6ab259b4df0fd5147bab0c949d81473db4518f81afc5c3f52f91ff6b34e",
           "n":0
          },
          "scriptSig":"3f3a4ce81...."
         }
        ],
        "out":[
         {
          "value":"10.12287097",
          "scriptPubKey":"OP_DUP OP_HASH160 69e02e18b5705a05dd6b28ed517716c894b3d42e OP_EQUALVERIFY OP_CHECKSIG"
         }
        ]
}
```

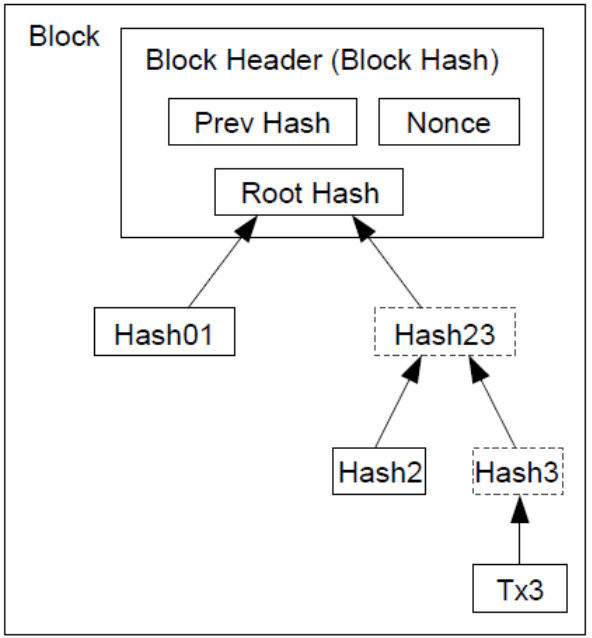

그림출처: Bitcoin and Cryptocurrency Technologies

2 블록의 구조

1) Merkle Trees



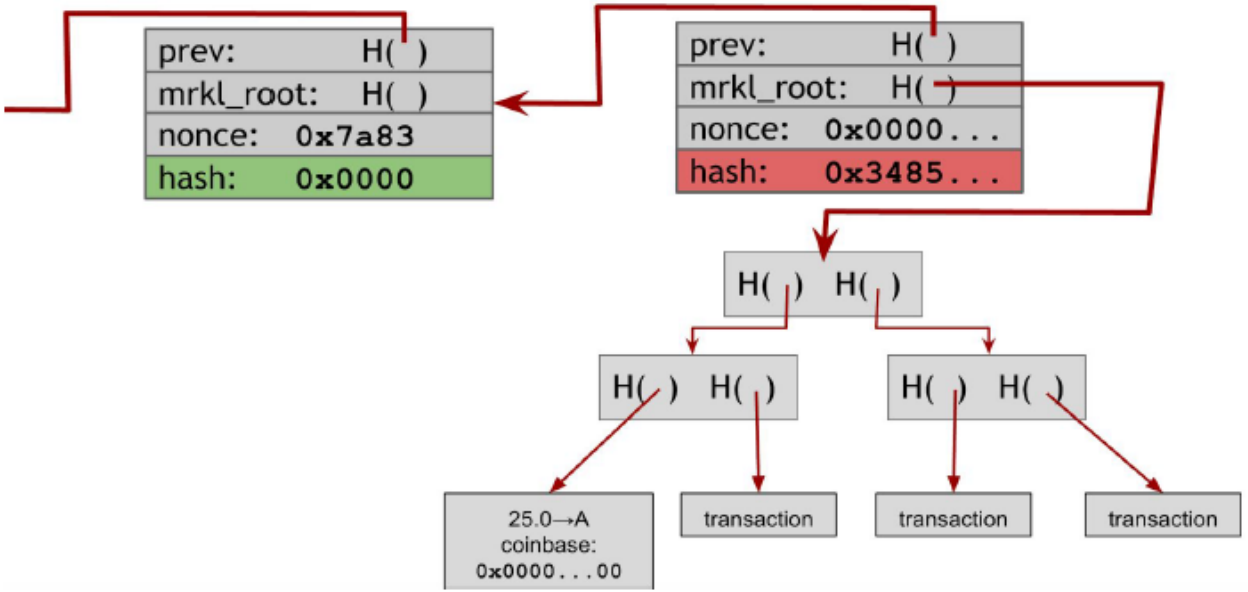
Transactions Hashed in a Merkle Tree



After Pruning Tx0-2 from the Block

그림출처: Bitcoin: A Peer-to-Peer Electronic Cash System

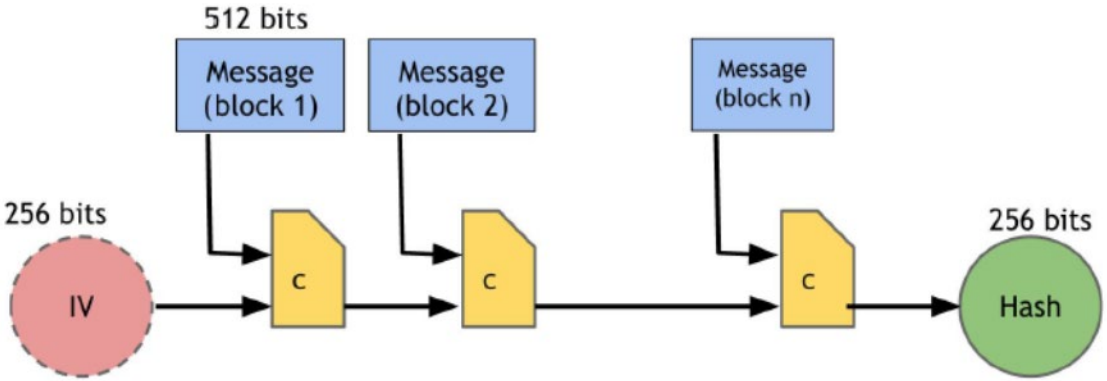
2) 비트코인 블록의 연결



그림출처: Bitcoin and Cryptocurrency Technologies

3) 비트코인 해시 함수 (SHA-256)

- 해시 함수의 input (any size)를 512 bits의 배수로 만듦
- Initial Vector(256 bits) + 512 bits → output (256 bits))
- 2^{256} 개의 가능한 해시 값 → collision resistant



그림출처: Bitcoin and Cryptocurrency Technologies

3 네트워크

- 다수의 노드로 구성된 P2P 네트워크
- 완전노드 (full node)
  - Bitcoin Core 프로그램
  - 모든 블록의 정보를 저장
  - 비트코인 채굴에 참여 가능
- 단순 지급 검증 노드 (simple payment node)
  - 블록들의 header 정보만 저장 (1년치 약 4.2MB)
  - 비트코인 지갑
- broadcasting, gossip protocol

4 작업증명 (Proof of Work)

1) 해시 퍼즐

- H(nonce ‖ prev_hash ‖ tx ‖ tx ‖ … ‖ tx)< target 을

만족하는 nonce 값을 찾아내면 블록이 생성 → 비트코인 채굴

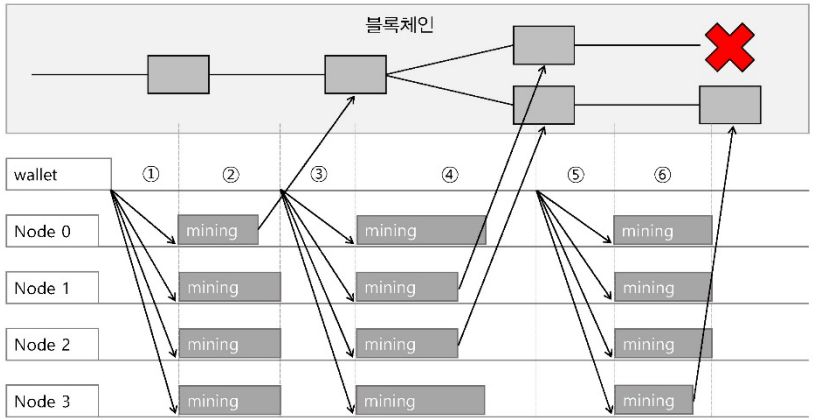
- 해시 퍼즐 난이도 를 나타내는 target 값 (16진수)

0000000000000000172EC0000000000000000000000000000000000000000000

- 전체 64개 자리 수 → $16^{64}=2^{256}$ 개의 가능한 해시 값
- 앞 부분 0의 자리 17개 → target < $16^{-17}=2^{-68}$
- 전력을 소비하여 단순계산을 하는 비효율적 방식

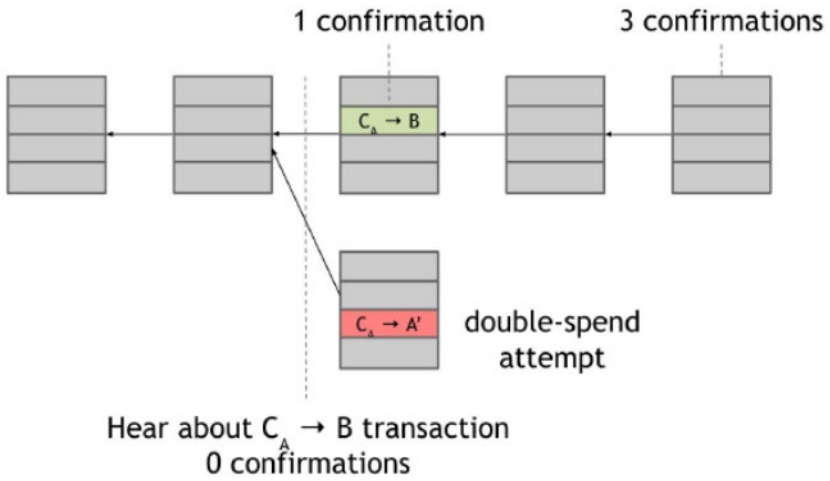
2) 작업증명 합의 알고리즘

- 먼저 해시 퍼즐을 푼 노드의 블록을 따라 새 블록 생성
- 블록 생성 후 이어서 6개의 블록이 더 생성되면 승인



3) 합의 알고리즘의 문제점

- 51% 공격: 절반을 초과하는 노드들이 같은 편
- 이중 지출 (double spending)

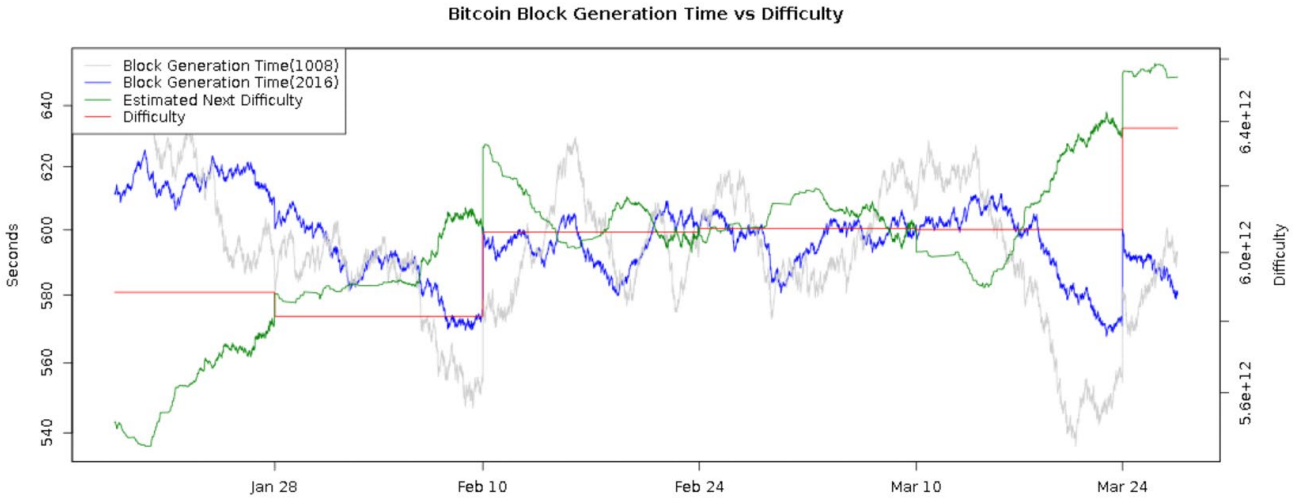


그림출처: Bitcoin and Cryptocurrency Technologies

4) 해시퍼즐 난이도 조절

› 매 2016 블록 (약 2주) 마다 조정

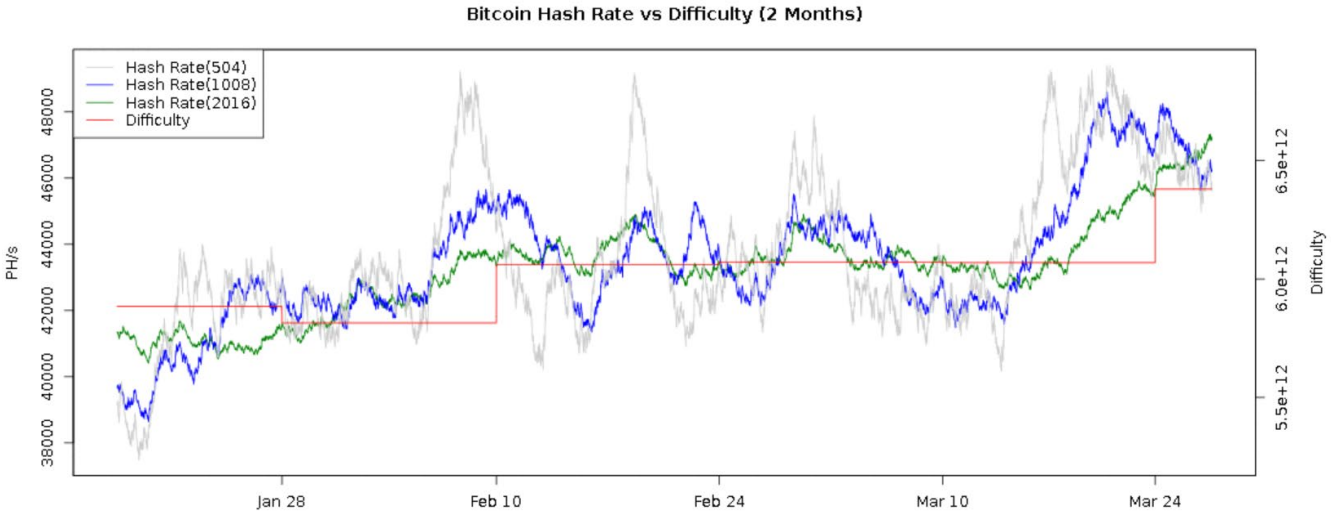
- 다음 난이도 = $\frac{\text{이전 난이도} * 2016 * 10\ (\text{분})}{\text{이전 2016 블록이 생성될 때 까지 걸린 시간}\ (\text{분})}$



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

5) 비트코인 hash rate 와 해시퍼즐 난이도

- hash rate 는 비트코인 채굴 역량을 의미
- hash rate 높아지면 작업증명 난이도 높아짐

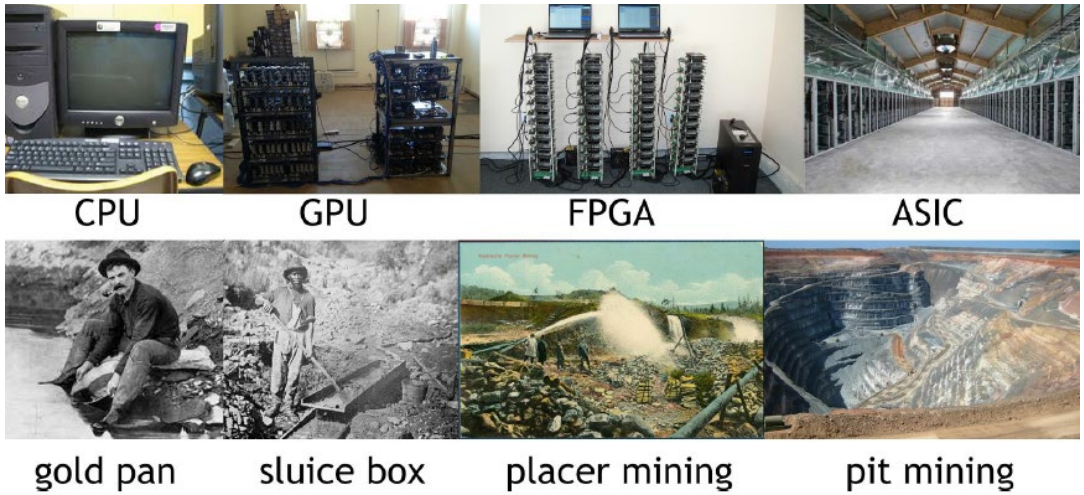


그림출처: https:// bitcoinwisdom.com/bitcoin/difficulty

5 비트코인 채굴

1) 채굴 하드웨어

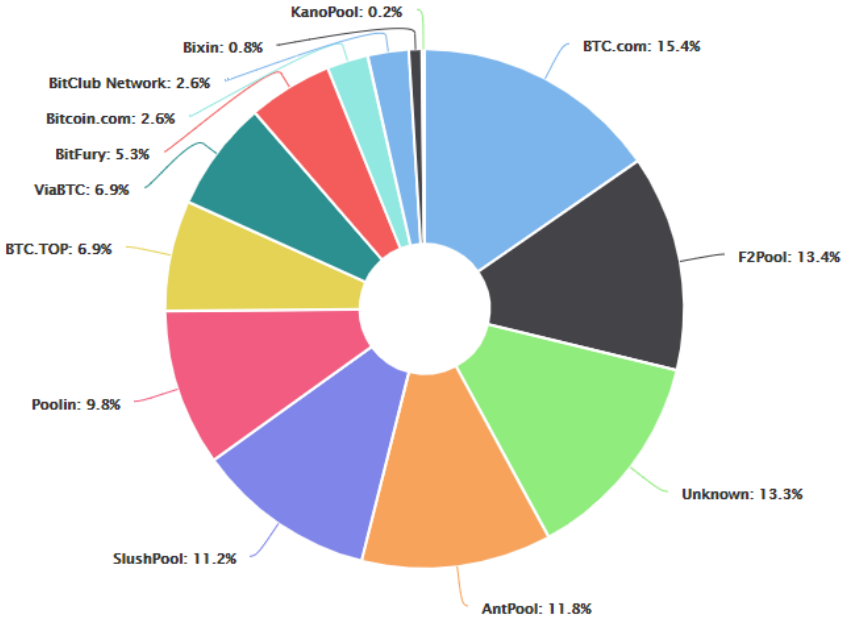
- 2018년 해시 퍼즐 난이도는 2009년에 비해 7조배 이상 상승
- 채굴 전용 기계 ASIC 의 끊임 없는 업그레이드



그림출처: Bitcoin and Cryptocurrency Technologies

2) Mining Pools

› 몇 안되는 마이닝 풀들이 비트코인 채굴 독점

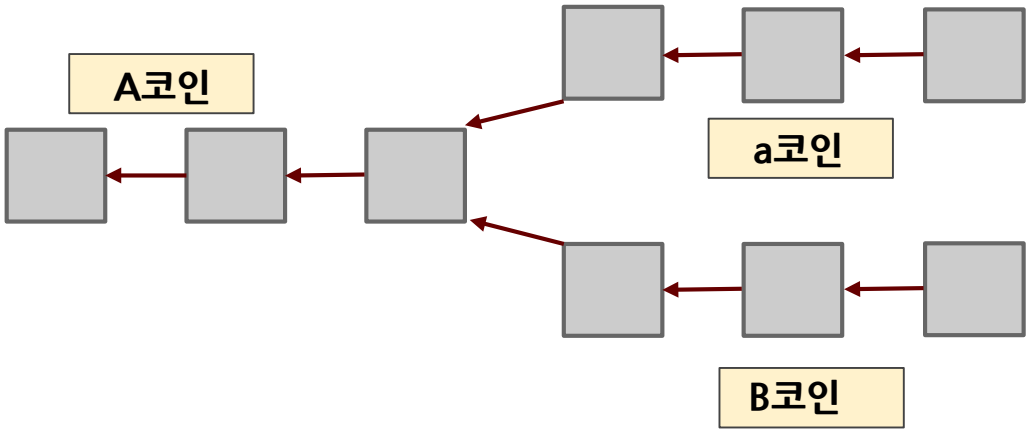


그림출처: https://www.blockchain.com

6 비트코인 포크

1) Currency Fork

- a 는 A 의 규칙을 따름 → 소프트 포크
- B 는 A 의 규칙을 따르지 않음 → 하드 포크



2) 소프트 포크 (soft folk)

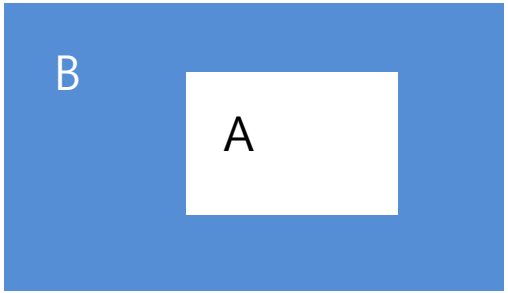
- SegWit (2017년 8월): 비트코인 블록에서

전자서명을 분리하여 비트코인 트랜잭션 저장공간 확대

- 기존 블록체인 A와 호환되는 개선된 블록체인 a

3) 하드 포크 (hard folk)

- Bitcoin Cash: SegWit 방식을 반대하는 하드포크
- Bitcoin Gold: 탈중앙화를 지향하는 하드포크
- 기존 블록체인 A와 호환되지 않는 개선된 블록체인 B



3. Idendity

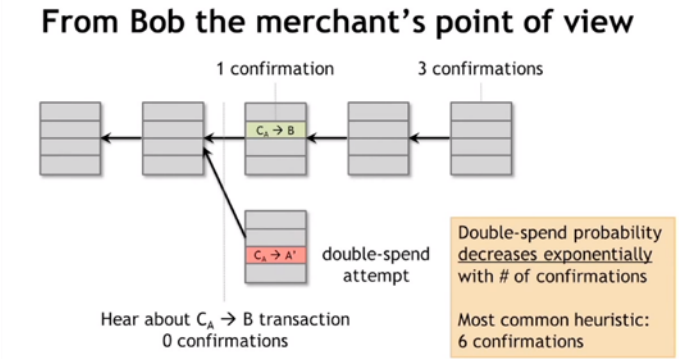
- 비트코인 절도
  해당 코인(공개키)을 소유한 자의 서명(비밀키)이 있어야 소유권을 변경할 수 있으므로
  암호화폐인 비트코인에서는 불가능

- 서비스 거부 공격
  서비스를 거부하지 않는 정직한 노드가 배정될 때 까지 기다리면 되므로 불가능

3. Idendity

- 이중지출 공격

  송금의 대상이 되는 상대방에게 신호를 보냄과 동시에 자신의 통제 하에 있는 다른 상대방에게 동일한 코인을 송금하는 공격 방식으로, 다음과 같이 일치 프로토콜에 의해 시간이 흐름에 따라 성공할 확률이 0에 기하급수적으로 수렴한다 :



1) 일반적으로 먼저 도착한 거래일수록 거래의 타당성이 더 많은 노드에 수용됨

(만약 거래 상대방이 거래 대상을 송금 이후에 지급한다면 사전에 이중지출 여부를 확인함)

2) 이중지출이 확인되지 않은 거래 건을 확인한 이후 거래를 완료함

3) 이후에 이중지출 공격을 시도하더라도 휴리스틱한 관점에서 노드들은 더 많이 채택된 거래를 자신의 블록에 포함시키므로 결국 이중지출된 거래가 유효할 가능성은 장기적으로 사장됨

* 실용적으로 볼 때 6개의 타당성 확인을 거친 거래는 확정적으로 유효성을 인정받게 된다고 알려짐

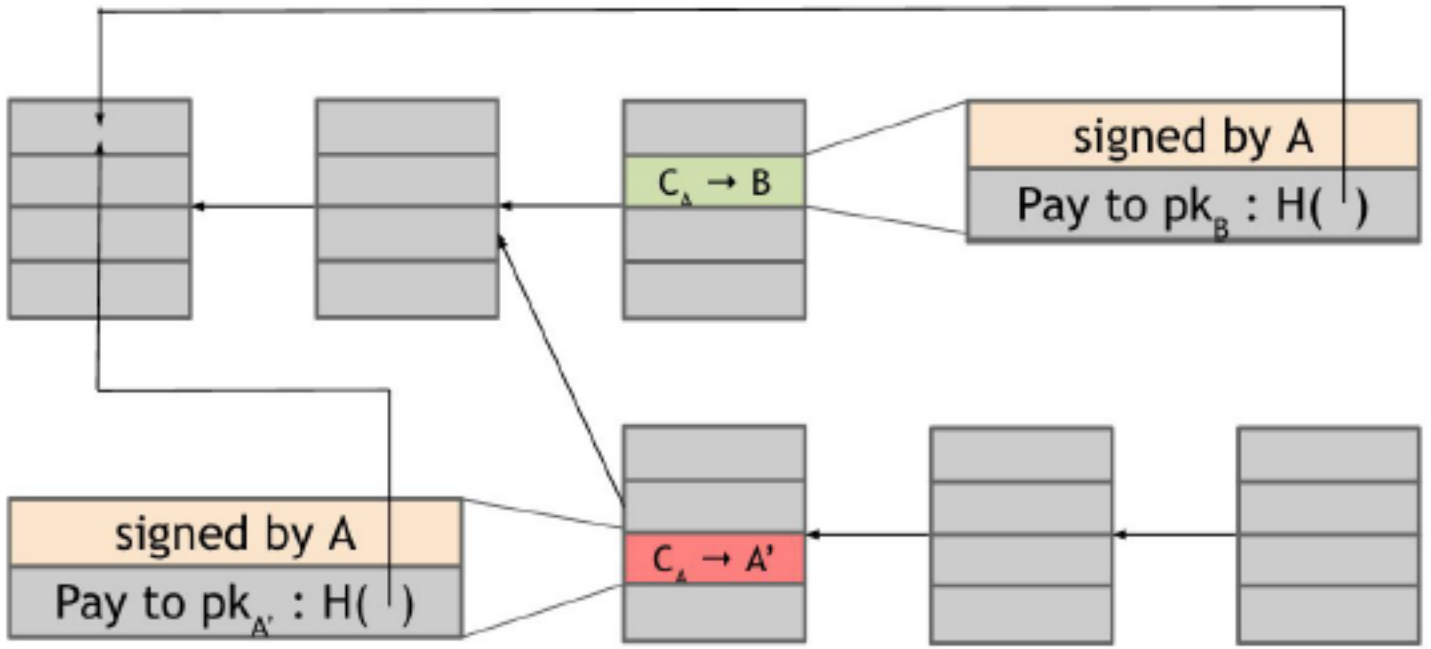


Figure 2.2 A double spend attempt. Alice creates two transactions: one in which she sends Bob Bitcoins, and a second in which she double spends those Bitcoins by sending them to a different address that she controls. As they spend the same Bitcoins, only one of these transactions can be included in the block chain. The arrows are pointers from one block to the previous block that it extends including a hash of that previous block within its own contents. C_A is used to denote a coin owned by Alice.

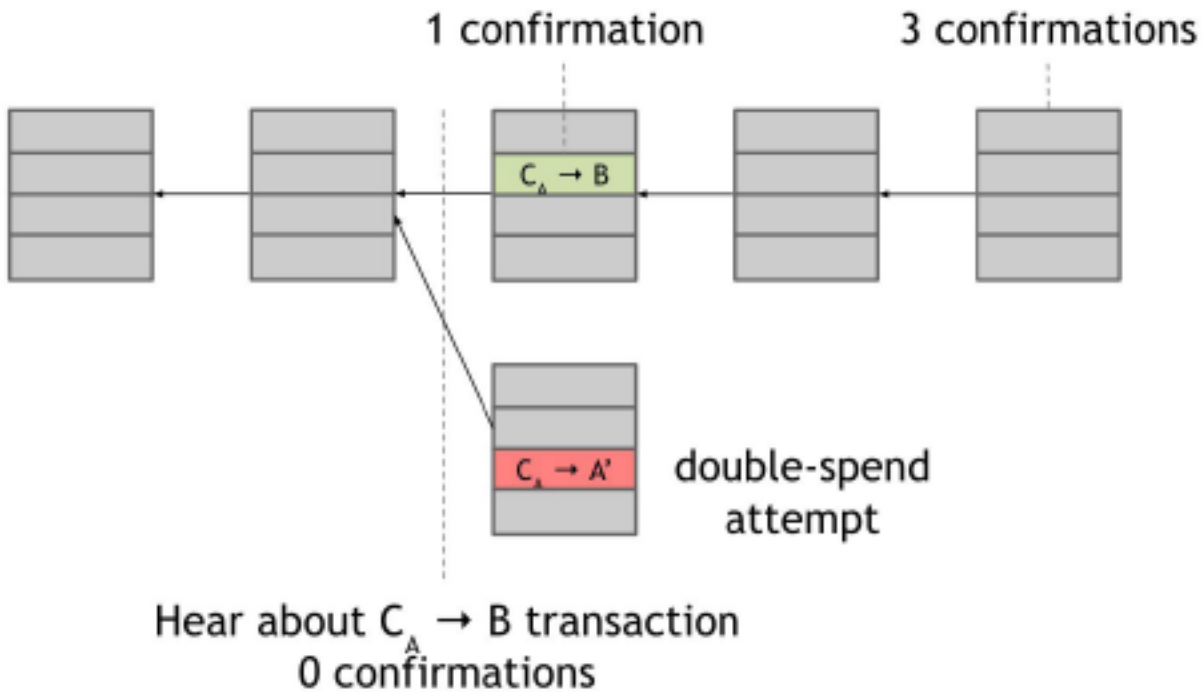


Figure 2.3 Bob the Merchant's view. This is what Alice's double-spend attempt looks like from Bob the merchant's viewpoint. In order to protect himself from this attack, Bob should wait until the transaction with which Alice pays him is included in the block chain and has several confirmations.

4. Block incentives

- ‘정직하지 못한’ 노드에 일일이 벌칙을 주는 것은 분산시스템에서는 실질적으로 불가능하다
  따라서 노드들이 정직하게 자발적으로 블록에 참여하고, 이에 따라 블록이 생성 및 유지되는 유인이 존재해야 된다
  (즉, 각 노드가 정직하게 행동하는 것이 내쉬 균형이 돼야함)

- 이와 같이 블록을 유지하는 데에는 두 가지 경제적 유인이 존재한다 :
  1) 블록 보상
  2) 거래 수수료

4. Block incentives

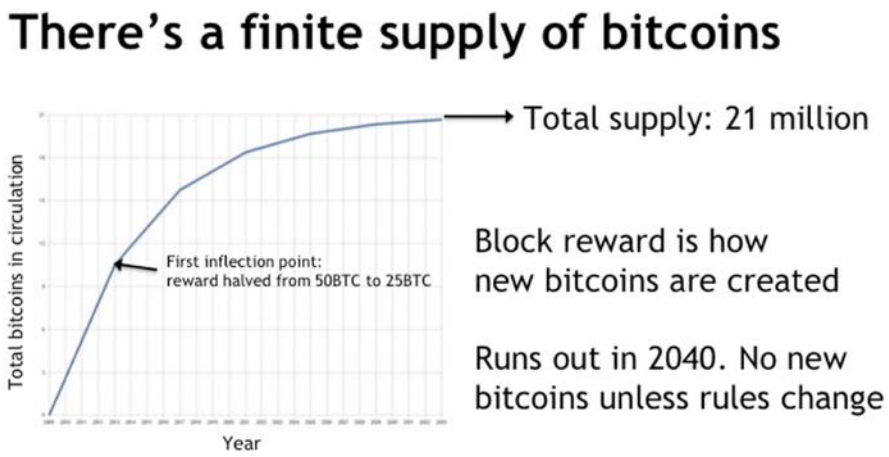
- **블록 보상**

  블록 생성자들은 **코인 생성**과, **생성된 코인의 주소를 배정**하는 데에 관여한다

  - 비트코인은 현재 1 블록 당 12.5 코인이 보상이며, 매 4년마다 보상은 반으로 줄어듦

  - 총 채굴량은 2100만 코인으로 한정돼있으며, 이에 따르면 2040년에 코인 생성과정은 중단됨

  이 때, 블록 생성자는 해당 블록이 장기적으로 '**일치**'된 체인의 말단에 속해야만 보상을 얻게 된다



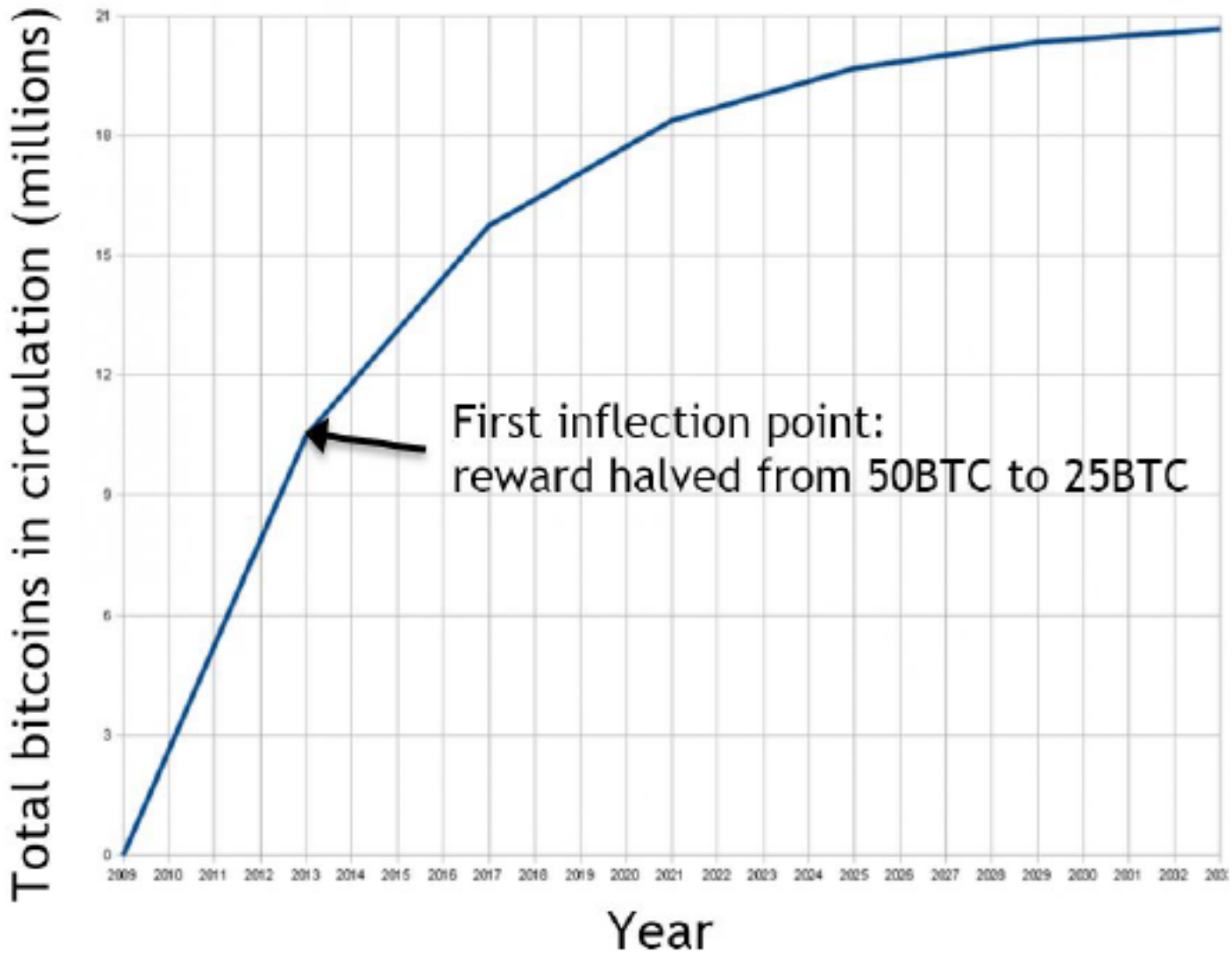


Figure 2.4 The block reward is cut in half every four years limiting the total supply of bitcoins to 21 million.

- 매 4년마다 채굴에 대한 보상으로 지급되는 BTC의 액수가 반감

2008 - 2012	50 BTC / 블록
2012 - 2016	25 BTC
2016 - 2020	12.5 BTC
2020 - 2024	6.25 BTC
⋮	⋮

- 매 4k (k=0, 1, 2, ...) 년 동안 $50\left(\frac{1}{2}\right)^k$ BTC / 블록이 지급
- 10분 마다 1개의 블록이 생성됨을 고려할 때, 4년 동안 생성되는 블록의 수

$$1\text{블록}/10\text{분} \times 60\text{분}/\text{시간} \times 24\text{시간}/\text{일} \times 365\text{일}/\text{년} \times 4\text{년} = 210,240\ \text{블록}$$

- 매 4k 년 동안 지급되는 BTC의 양 $10,512,000\left(\frac{1}{2}\right)^k$ BTC
- 총 채굴되는 BTC의 양 $10,512,000\sum_{k=0}^{\infty}\left(\frac{1}{2}\right)^k = 21,024,000$ BTC

$$10,512,000\sum_{k=0}^{\infty}\left(\frac{1}{2}\right)^k \approx 10,512,000\sum_{k=0}^{7}\left(\frac{1}{2}\right)^k \approx 2.1\times10^6\ \text{BTC}$$

2012 + 4×7 = 2040년 무렵 채굴 거의 끝남

4. Block incentives

- **거래수수료**

  거래 생성자들은 자발적으로 블록 생성자에게 거래 수수료를 남겨야 한다

  현 시스템 상 강제는 아니지만, 먼 미래에 코인 생성이 중단되는 시점에서는 시스템(비트코인 소프트웨어)가 이를 바꿀 수도 있음

If

mining reward > mining cost

then miner profits

where

mining reward = block reward + tx fees

mining cost = hardware cost + operating costs (electricity, cooling, etc.)

4. Block incentives

- 초기에 제시했던 일련의 과정에 대해 다음과 같은 몇 가지 문제가 남는다 :

  1) 임의 노드를 어떻게 뽑는가?

  2) 보상을 위한 무한경쟁이 방지되는 메커니즘은 무엇인가?

  3) 시빌 공격을 어떻게 막는가?

  이 모든 문제의 해답은 **작업증명 (Proof of Work)** 이라는 개념에 의해 설명할 수 있다

  Hash Puzzle $H(nonce \| prev\_hash \| tx \| tx \| \ldots \| tx) < target$

  임의 노드를 뽑기 위해 아무도 독점할 수 없을 만큼의 자원의 비율로 노드를 선택하는데,

  해시 연산력의 비중에 따라 분배하는 **작업증명**과 코인의 소유에 따라 분배하는 **자산증명 (Proof of Status)**이 있다

계좌 기반 일반적인 장부

Create 25 coins and credit to Alice$_{\text{ASSERTED BY MINERS}}$
Transfer 17 coins from Alice to Bob$_{\text{SIGNED(Alice)}}$
Transfer 8 coins from Bob to Carol$_{\text{SIGNED(Bob)}}$
Transfer 5 coins from Carol to Alice$_{\text{SIGNED(Carol)}}$
Transfer 15 coins from Alice to David$_{\text{SIGNED(Alice)}}$

Figure 3.1 an account-based ledger

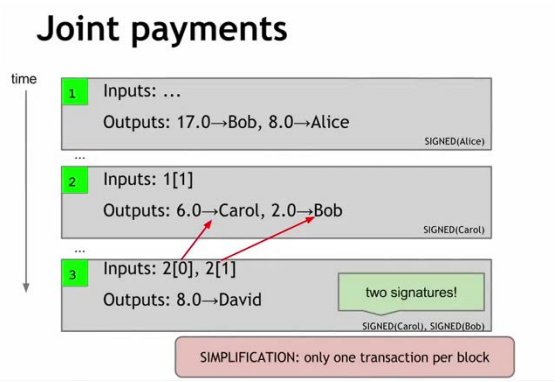
거래 기반 비트코인 장부

1	Inputs: Ø Outputs: 25.0→Alice	
2	Inputs: 1[0] Outputs: 17.0→Bob, 8.0→Alice	SIGNED(Alice)
3	Inputs: 2[0] Outputs: 8.0→Carol, 9.0→Bob	SIGNED(Bob)
4	Inputs: 2[1] Outputs: 6.0→David, 2.0→Alice	SIGNED(Alice)

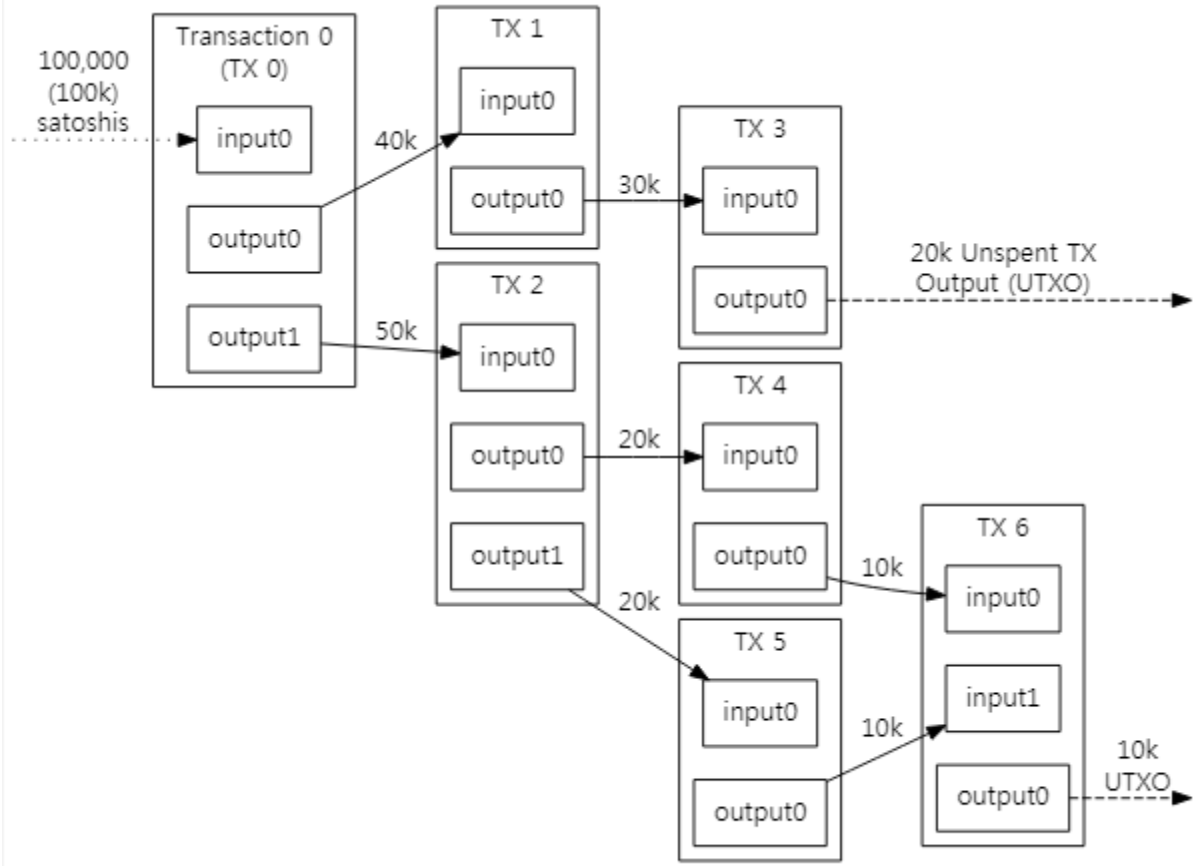
Figure 3.2 a transaction-based ledger, which is very close to Bitcoin

- 계좌 기반 장부를 보면 마지막 거래를 하기 위하여 모든 계좌의 내용을 추적 해야 하므로 계좌의 크기에 대한 문제와 자금 추적의 어려움이 있다.

- 거래 기반 장부는 이전 거래를 추적하여 해당 거래가 valid한지 검증할 수 있다.



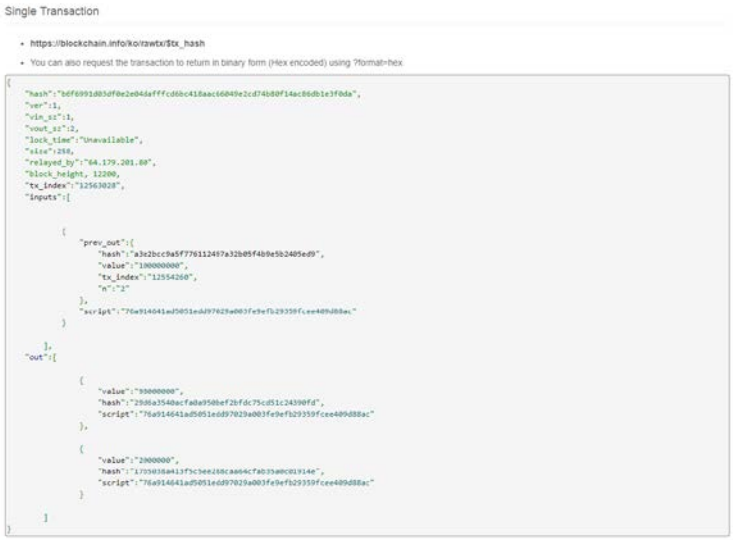
- 또한, 한번의 거래에 input이 여러 개, output이 여러 개 존재할 수 있다.



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin

비트코인 거래의 스크립트 예시

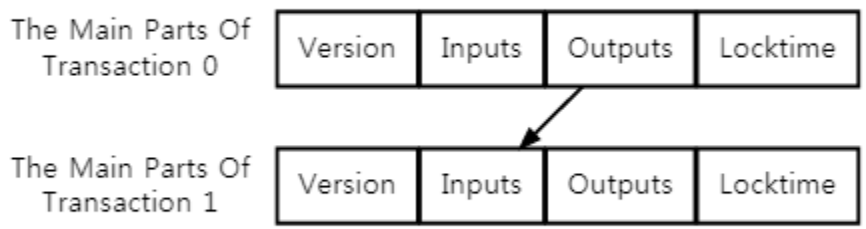
Blockchain.info



- 실제 비트코인 거래는 metadata, input, output으로 구성 되어 있는다.
- Metadata에는 해당 거래의 해쉬와 거래의 크기, input 개수, lock_time 등으로 구성
- Input에는 이전 거래의 해쉬와 비트코인을 보내는 사람의 script signature으로 구성
- Output에는 비트코인의 가치와 script public key으로 구성

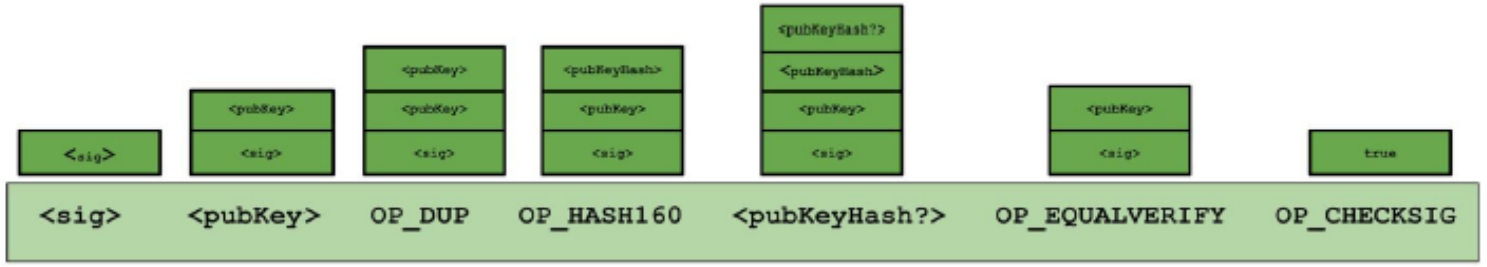
Figure 3.3 **An actual Bitcoin transaction.**

Each input spends a previous output

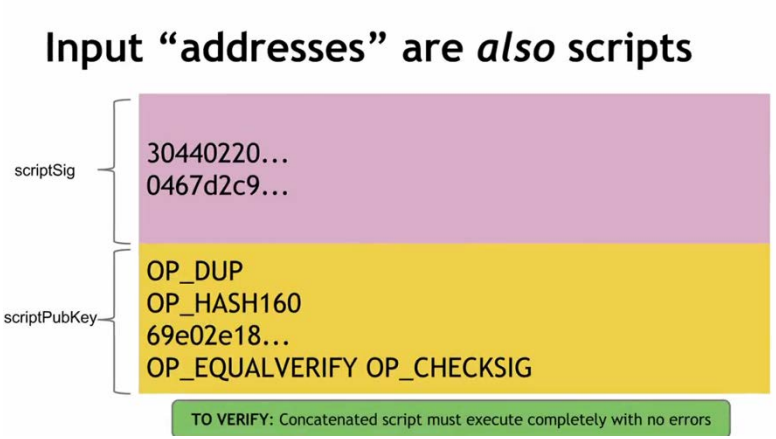


Each output waits as an Unspent TX Output (UTXO) until a later input spends it

비트코인 거래 스크립트에서 사용하는 언어



- 비트코인 거래의 스크립트에서 사용하는 언어는 stack-based로 간단하고 간결, LIFO방식, loop가 일어나지 않는다.
- 해당 거래가 valid한지 확인하기 위하여 Script를 이용하는데, 해당 거래에 input에 들어가 있는 Script Signature과 이를 참조하고 있는 거래의 Output Script Public key를 이용하여 거래가 Valid한지 검증한다.



비트코인 거래 스크립트에서 사용하는 언어

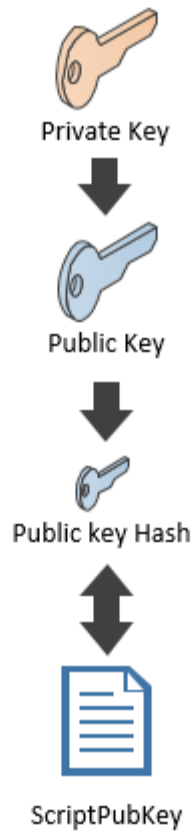
`OP_DUP`	Duplicates the top item on the stack
`OP_HASH160`	Hashes twice: first using SHA-256 and then RIPEMD-160
`OP_EQUALVERIFY`	Returns true if the inputs are equal. Returns false and marks the transaction as invalid if they are unequal
`OP_CHECKSIG`	Checks that the input signature is a valid signature using the input public key for the hash of the current transaction
`OP_CHECKMULTISIG`	Checks that the *k* signatures on the transaction are valid signatures from *k* of the specified public keys.

Figure 3.6 a list of common Script instructions and their functionality.

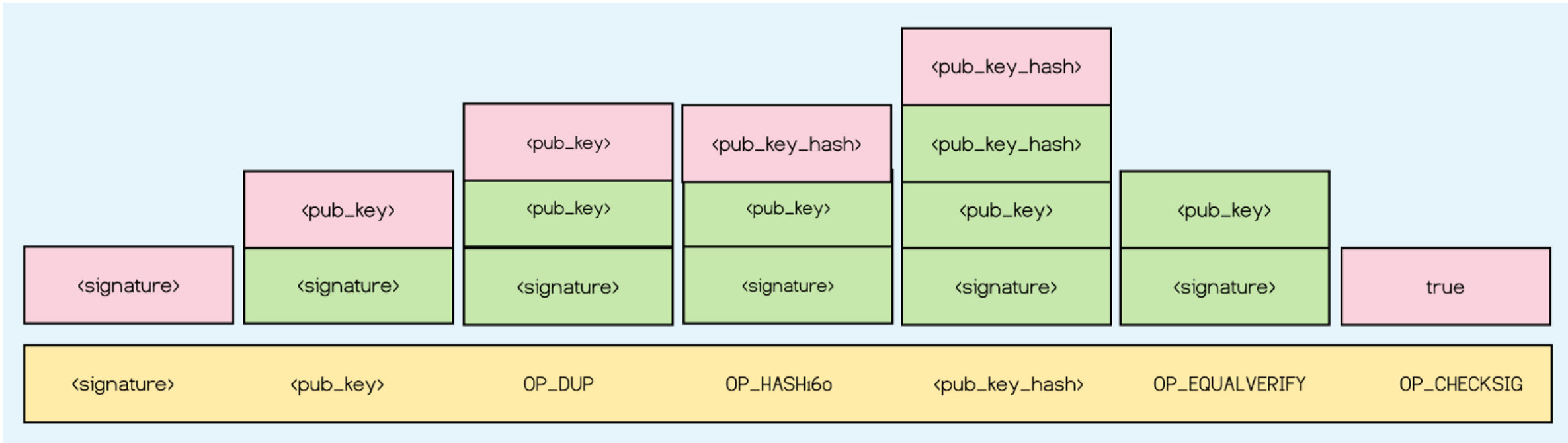
- Bitcoin Script에서는 75개의 instruction이 존재하는데 그 중 거래가 valid한 지 검증할 때 쓰이는 지시어이다.

- DUP은 가장 위 stack 복사
- Hash160은 가장 위 stack 해쉬 값 도출
- EQUALVERIFY는 가장 위 두 stack이 일치하는지 확인
- CHECKSIG는 public key를 이용하여 signature가 valid하는지 확인하는 지시어
- CHECKMULTISIG는 여러 개 public key에 대한 여러 signature 확인

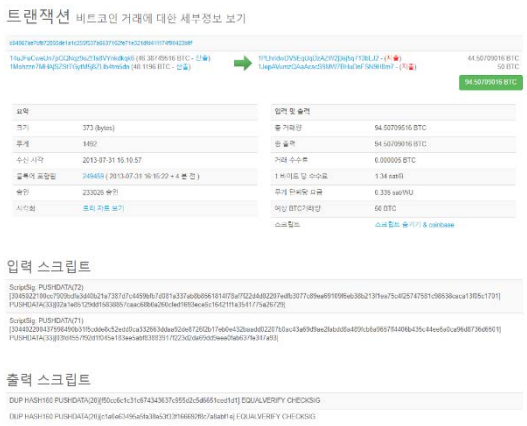
Pay To Public Key

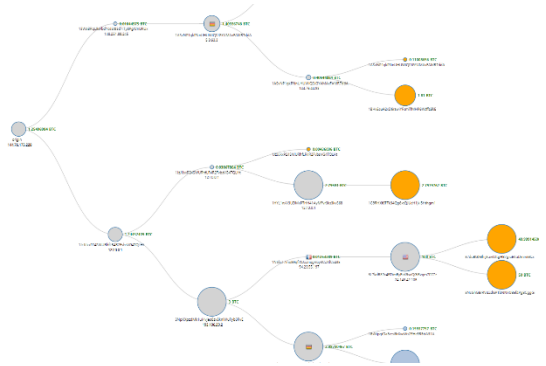


Pay To Public Key Hash


<signature>
<pub_key>
<signature>
<pub_key>
<signature>
<pub_key_hash>
<pub_key>
<signature>
<pub_key_hash>
<pub_key_hash>
<pub_key>
<signature>
<pub_key>
<signature>
true
<signature>
<pub_key>
OP_DUP
OP_HASH160
<pub_key_hash>
OP_EQUALVERIFY
OP_CHECKSIG

비트코인 거래 Blockchain info





- Proof of burn -> 다시 코인을 사용 할 수 없는 스크립트로 OP_RETURN이라는 지시어를 이용하여 만든다.(alt coin을 만들 때 사용하는 거 같은데 자세한 건 10장..)

- P2SH -> 일반적으로는 P2PKH로 public key를 코인을 보내는 주소로 사용하는데, Multi signature을 이용하지만 복잡하지 않고 간단하게 거래를 하기 위하여 Script에 보내는 경우가 있다.