Bitcoin

50.037 Blockchain Technology

Paweł Szałachowski

Bitcoin

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

- White paper, 2008
- Open and distributed consensus (with incentives and no trusted parties)
- Peer-to-peer (equal participants)
- Cheap, anonymous/private, instant, and censorship-free payments
- Bitcoins (₿)
  - current supply: 17M₿
  - total supply: 21M₿
  - 100,000,000 Satoshi = 1₿

Bitcoin

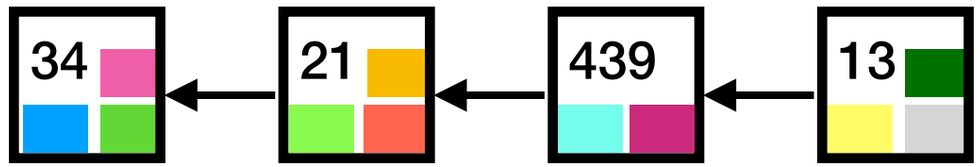
- Previous systems
  - Ecash (D. Chaum)
  - Cypherpunks
- Issues
  - Centralization and Trust
  - Lack of incentives
  - Security

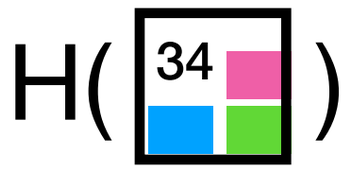
- **Bitcoin** (Core)
- Bitcoin Cash
- Bitcoin Gold
- Bitcoin Diamond
- Bitcoin Private
- … hundreds forks …

Recap & High-level Overview

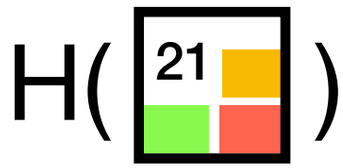
Bitcoin’s Blockchain

- Hash chain of blocks
- Block
  - Set of transactions
    - Forming hash tree
  - Special number (Nonce)
  - Hash pointer/link to the previous block
- Proof of work
  - H(block) < target (target defines difficulty)
    - You can see leading zeros in hashes
  - H(.) defined as SHA256(SHA256(.))

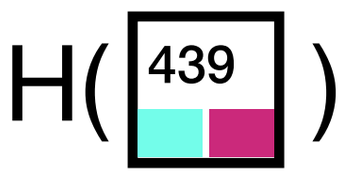




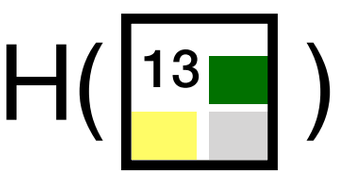
00000000892fcb17…



00000000490747db…



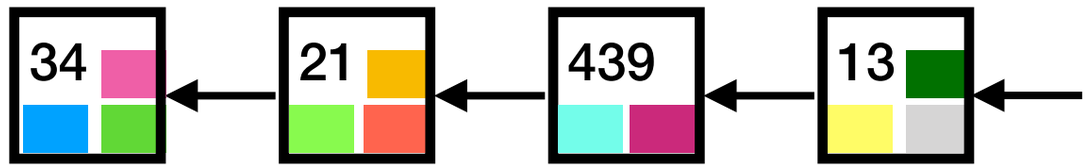
00000000a4d587e0…



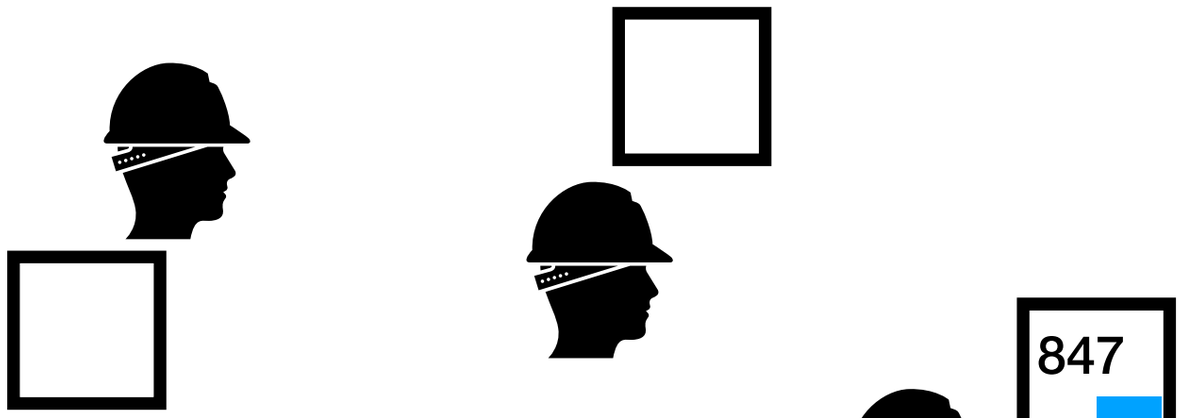
00000000ff410ef45…

Transactions and Mining

Blockchain



Miners
(Bitcoin network)



847

+12.5 BTC

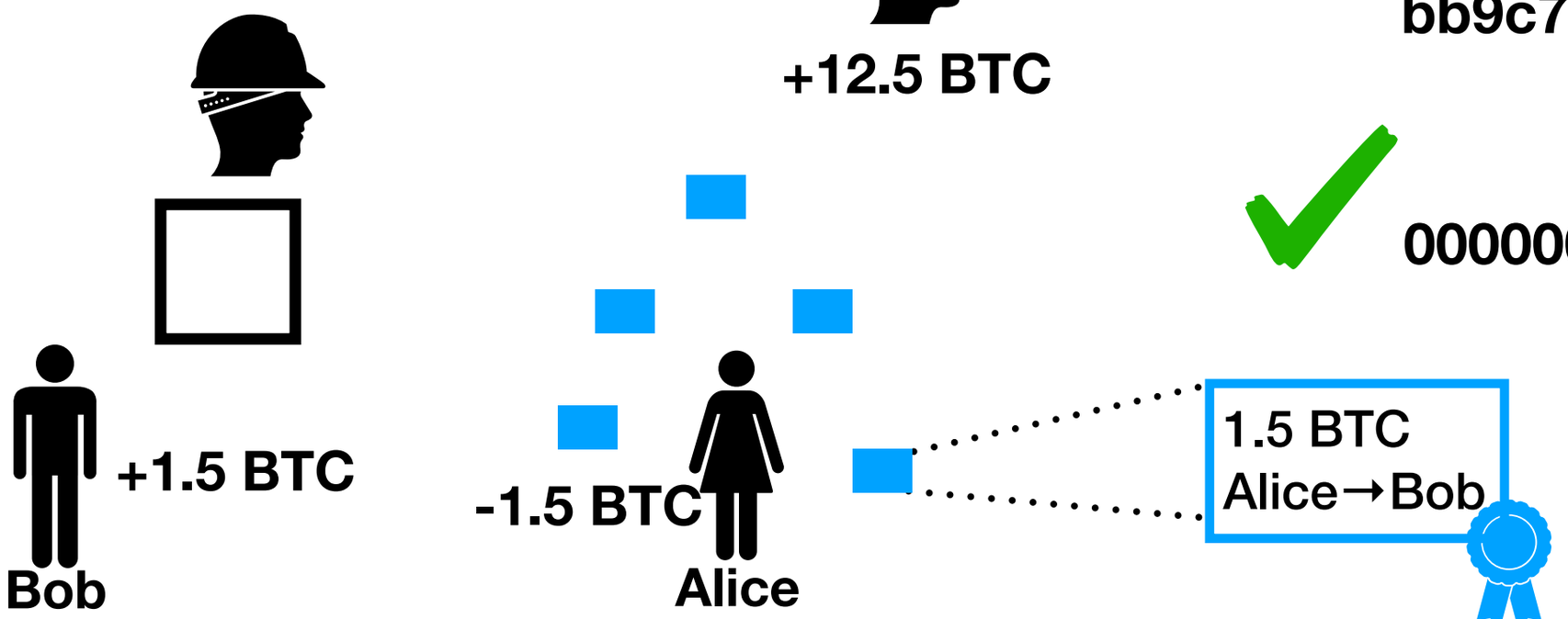
f54ee8e79bad... H(1)

37f082290f4f7... H(2)

bb9c721da24... H(3)

...

000000000c5... H(847)

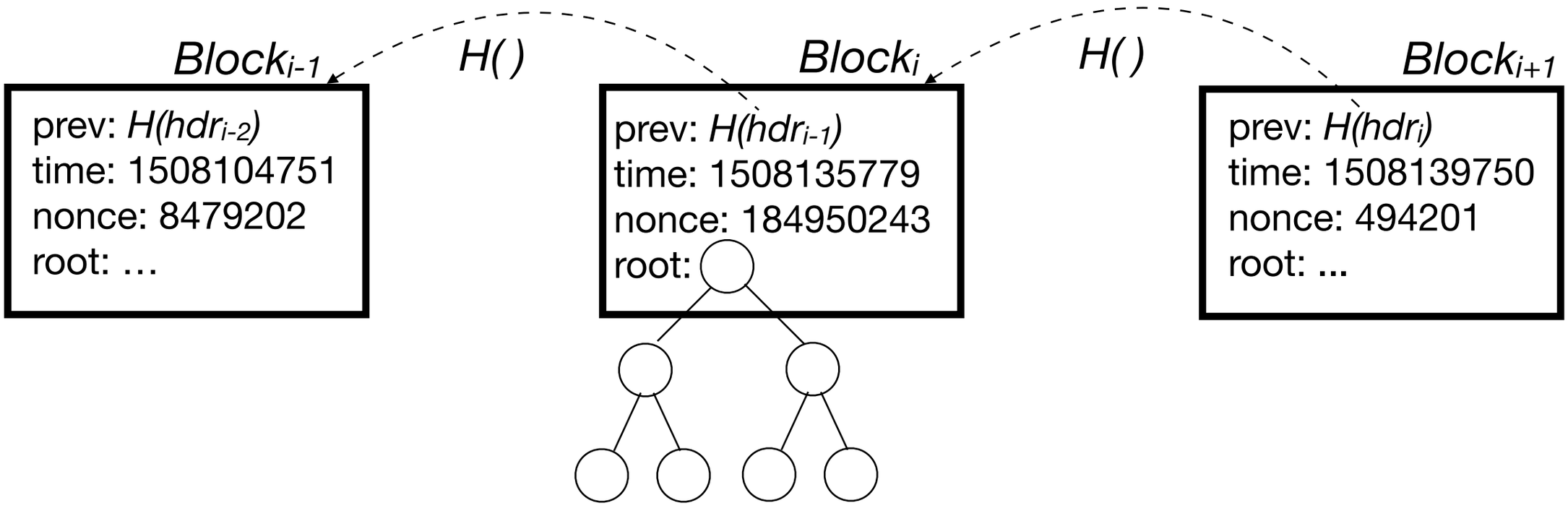


Low-level Overview

Block and Header

- Block
  - Block header
  - Transactions (form a hash tree)

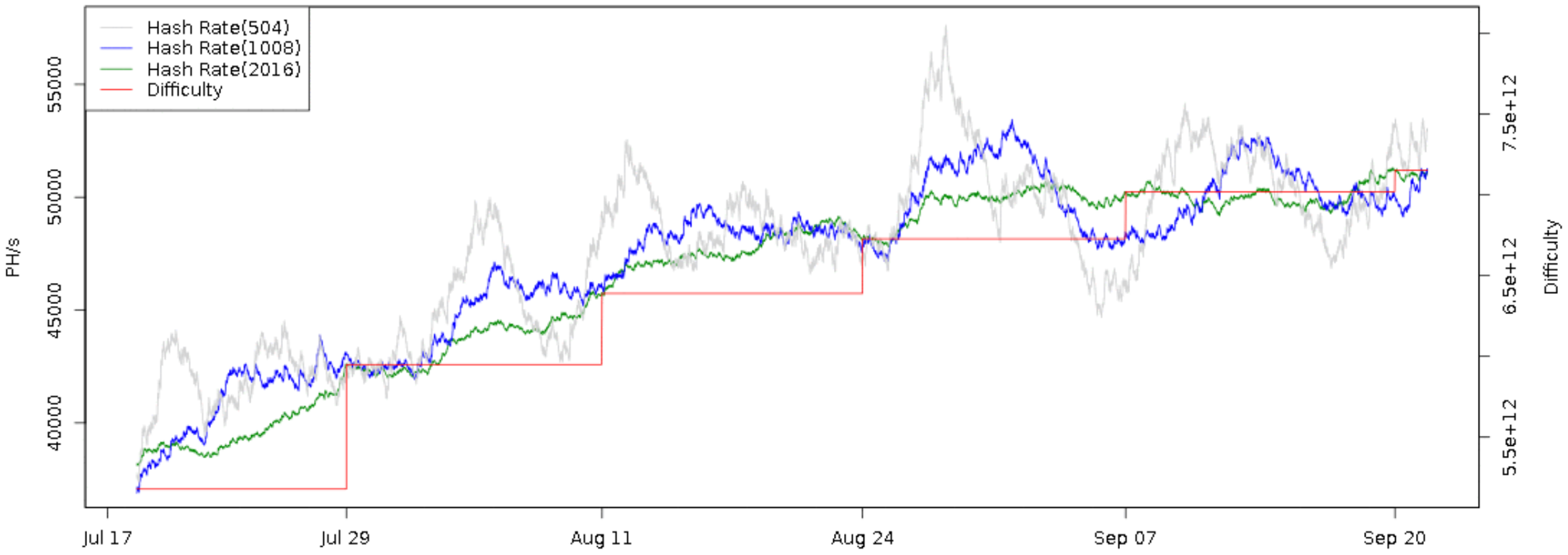
Field	Purpose
Version	Block version number
hashPrevBlock	256-bit hash of the previous block header
hashMerkleRoot	256-bit hash based on all of the transactions in the block
Time	Current timestamp as seconds since 1970-01-01T00:00 UTC
Bits	Current target in compact format
Nonce	32-bit number (starts at 0)



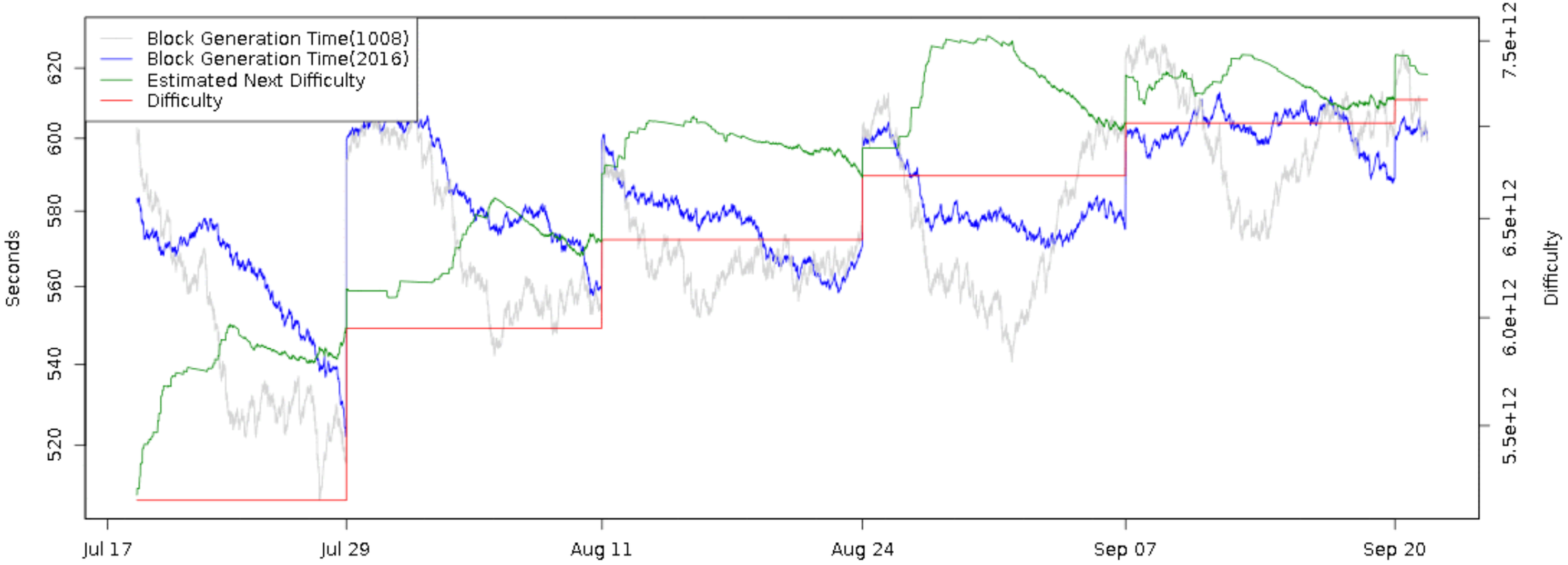
Proof of Work

- Nakamoto Consensus maintained by anonymous peers
  - Hash of block header < **target** (proves that a lot of PoW was invested)
  - Security (to misbehave a significant mining power is required)
  - Fast verification
- Target & Difficulty
  - **difficulty** = constant / current_target
  - Current target: 00000000000000000121ad4000000000000000000000000000000…
- Difficulty Adjustment (every 2016 blocks, the ideal value is two weeks, i.e., 10 min/block)
  - $\Delta = B_{(i+1)*2016}.\text{timestamp} - B_{i*2016}.\text{timestamp}$
  - If (Δ < two weeks) then increase difficulty proportionally (max 300%)
  - If (Δ > two weeks) then decrease difficulty proportionally (max 75%)

Bitcoin Hash Rate vs Difficulty (2 Months)



Bitcoin Block Generation Time vs Difficulty



Block and Header Validation

- Miners propagate blocks
- A new block is accepted if
  - Size <= MAX_BLOCK_SIZE (1MB, very controversial topic…)
  - Hash of its header < current target
  - The previous block’s hash is correct
  - Transactions form a hash tree whose root is in the header
  - All its transactions are *correct*
  - The timestamps does not deviate from the network time
  - …

Full Nodes

- Full Nodes
  - Replicated state machine
    - **Share state** managed by replicas
      - Set of unspent coins
    - Apply **operations** to modify it over time
      - Transactions
      - Receive, validate, and store all blocks
- Miners are full nodes running the consensus protocol
  - Propagate new blocks

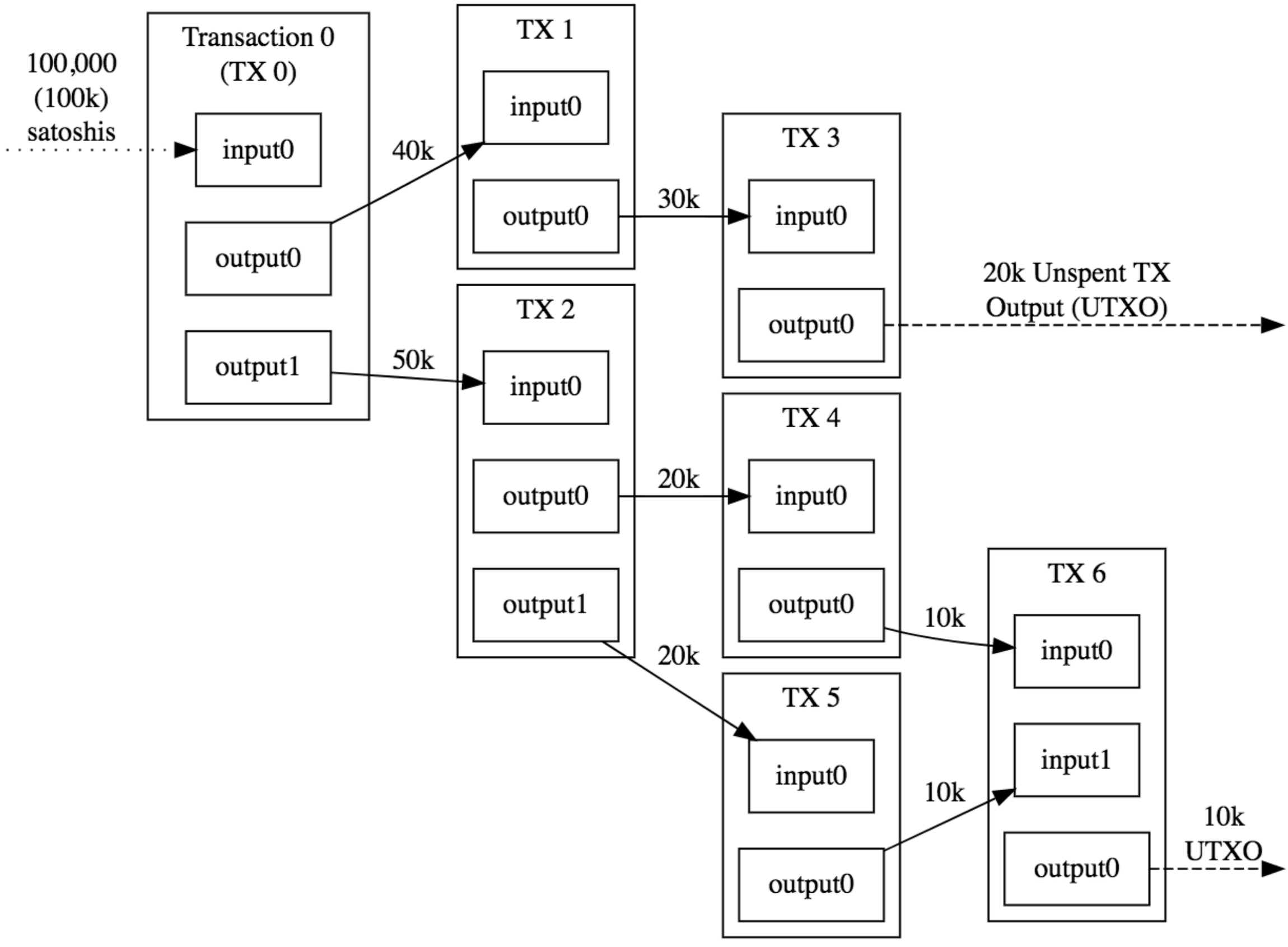
Transactions and the UTXO Model

Transaction Data

- Transaction
  - Sender (coins to be spent)
  - Receiver (how to spend them later)
  - Fee (incentivize miners)
- One or more transaction per block
- Transactions are identified by their hashes
  - TXID = H(transaction)

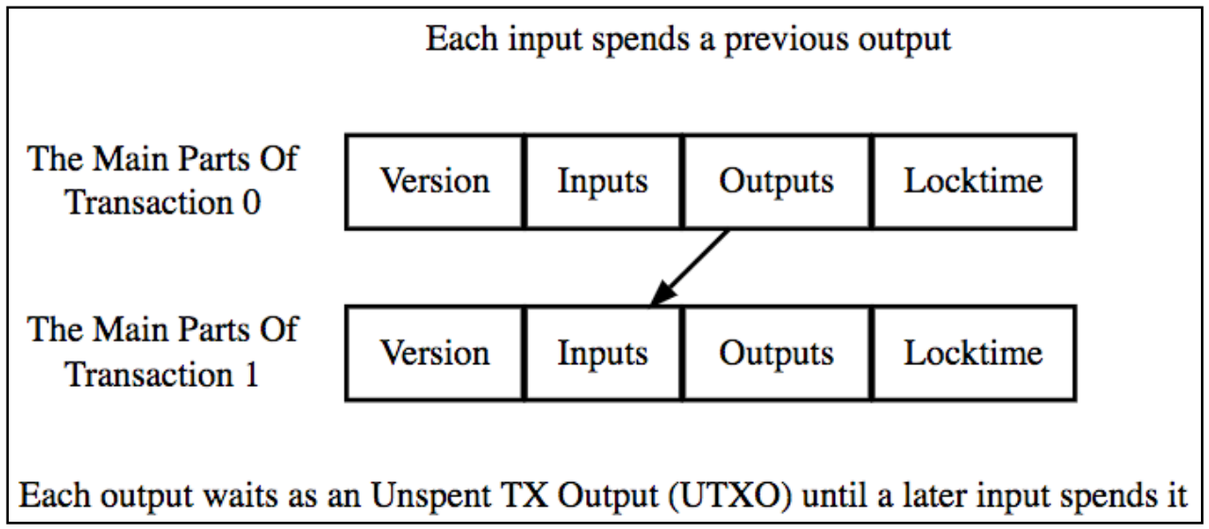
UTXO Transaction Model

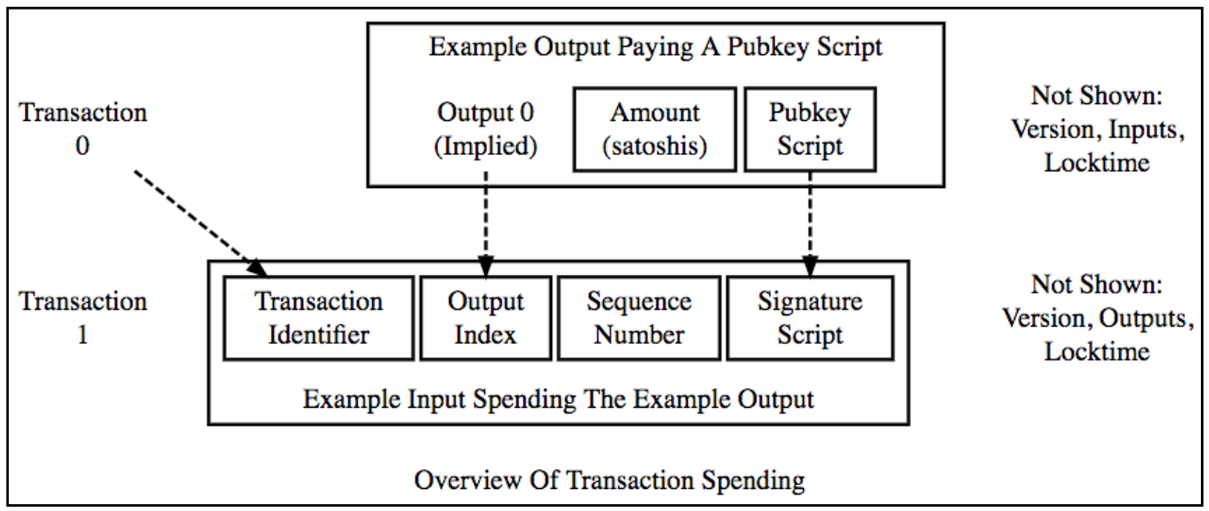
- The Unspent Transaction Output (UTXO) Model
- Each transaction spends the coins previously received in one or more earlier transactions
    - The input of one transaction is the output of a previous transaction
- A single transaction can create multiple outputs
    - Each output can only be used as an input once
- Transaction validity
    - Must only use UTXOs as inputs
    - Value of its outputs <= inputs
        - The difference (inputs - outputs) can be claimed as a fee by the miner who creates the block containing this transaction


100,000
(100k)
satoshis
Transaction 0
(TX 0)
input0
output0
output1
40k
50k
TX 1
input0
output0
TX 2
input0
output0
output1
20k
20k
30k
TX 3
input0
output0
TX 4
input0
output0
20k Unspent TX
Output (UTXO)
10k
10k
TX 5
input0
output0
TX 6
input0
input1
output0
10k
UTXO

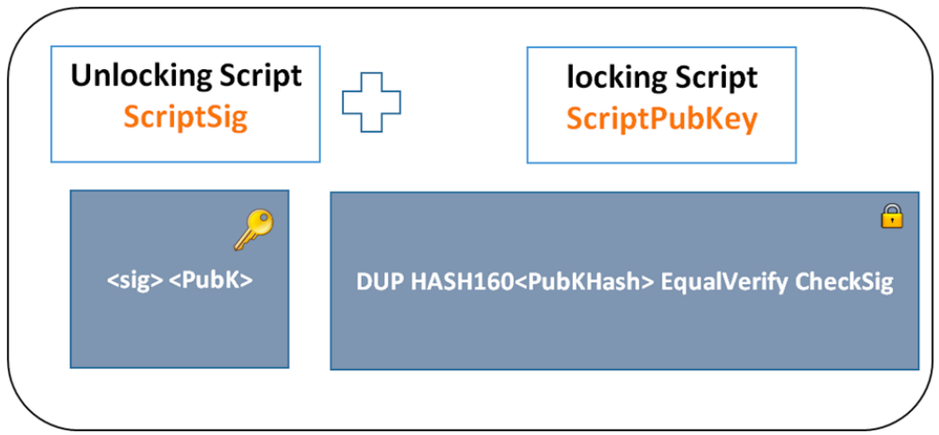
Transaction Format

- Version determines set of rules to apply
- An input uses a TXID and an output index number to identify a particular output to be spent
- It also has a **signature script** which allows it to provide data parameters that satisfy the conditions (**pubkey script**) to spend the output
- Locktime
  - The earliest time a transaction can be added to the blockchain
  - New non-locked transaction can invalidate it (use the same outputs



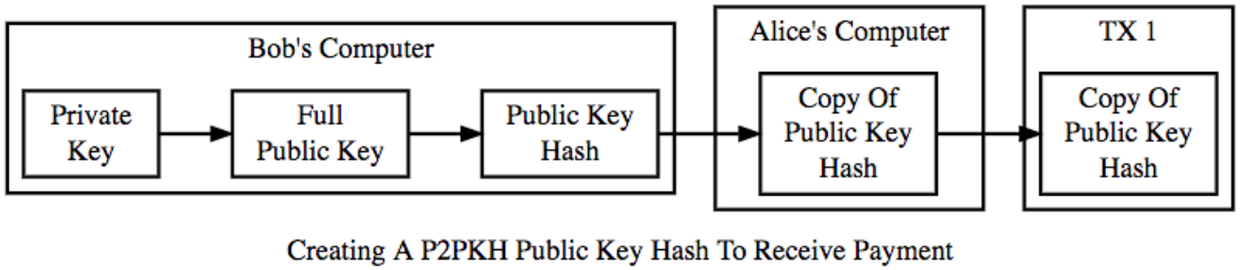


Pay-To-Public-Key-Hash (P2PKH)

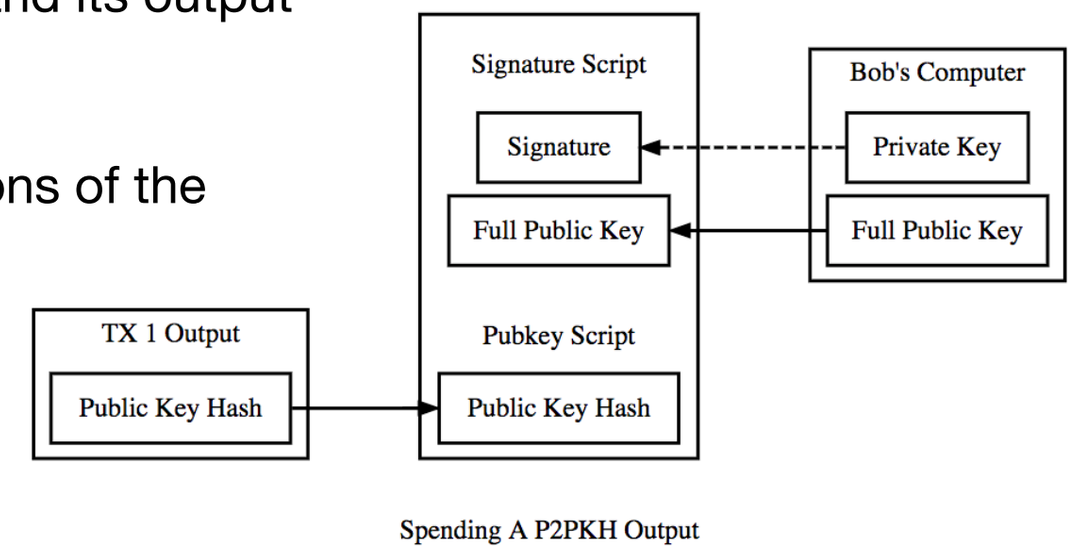


src: cryptocompare.com

- Digital signatures (public/private keys)
- Alice sends transaction to Bob's address
  - Address is encoded as public key's hash (base58)
  - P2PKH transaction output that allows to spend it anyone who can prove its ownership of the private key corresponding to Bob's hashed public key
    - These instructions are called the **pubkey script** or scriptPubKey
- Transaction is propagated and Bob treats it as a spendable balance
- Bob spends it creating an input referring to the Alice's TXID and its output index
  - and creating a **signature script** that satisfies the conditions of the output's pubkey script



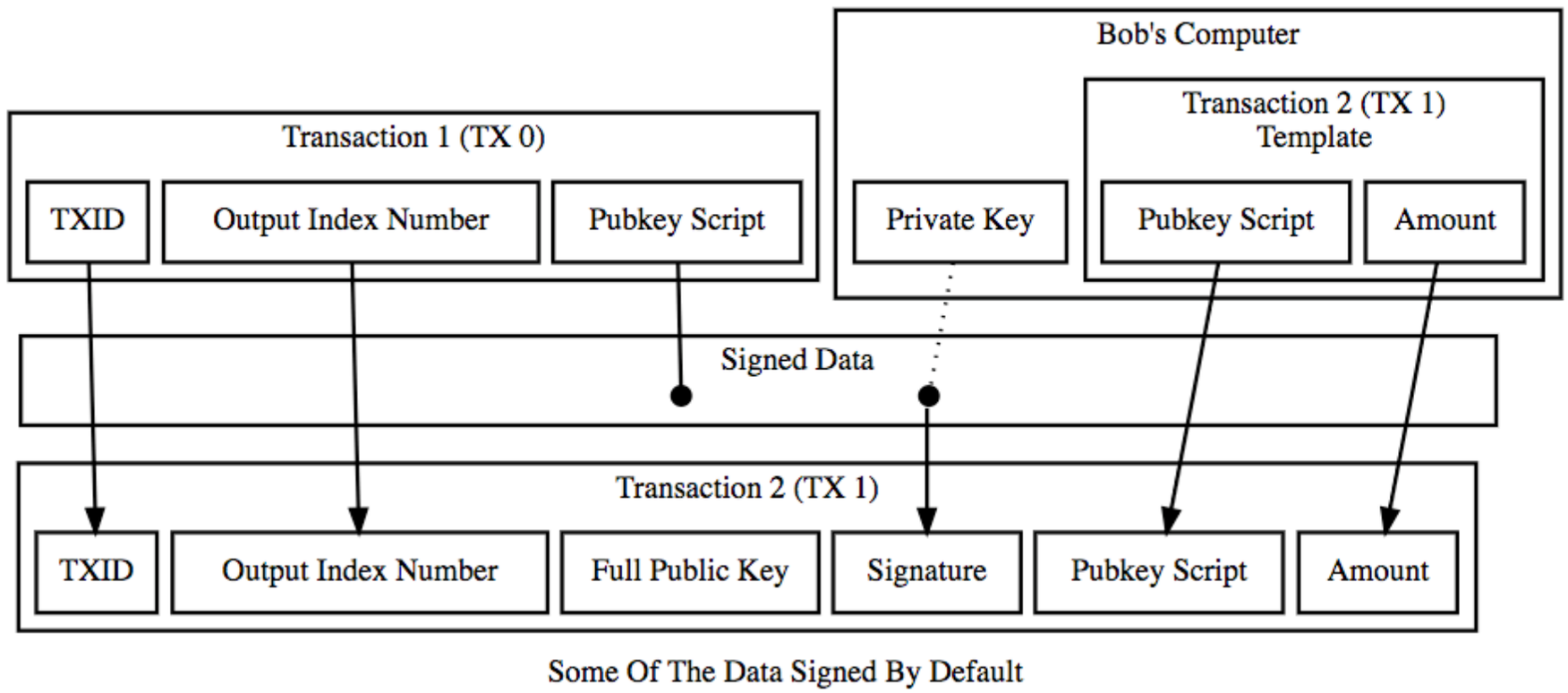
Creating A P2PKH Public Key Hash To Receive Payment



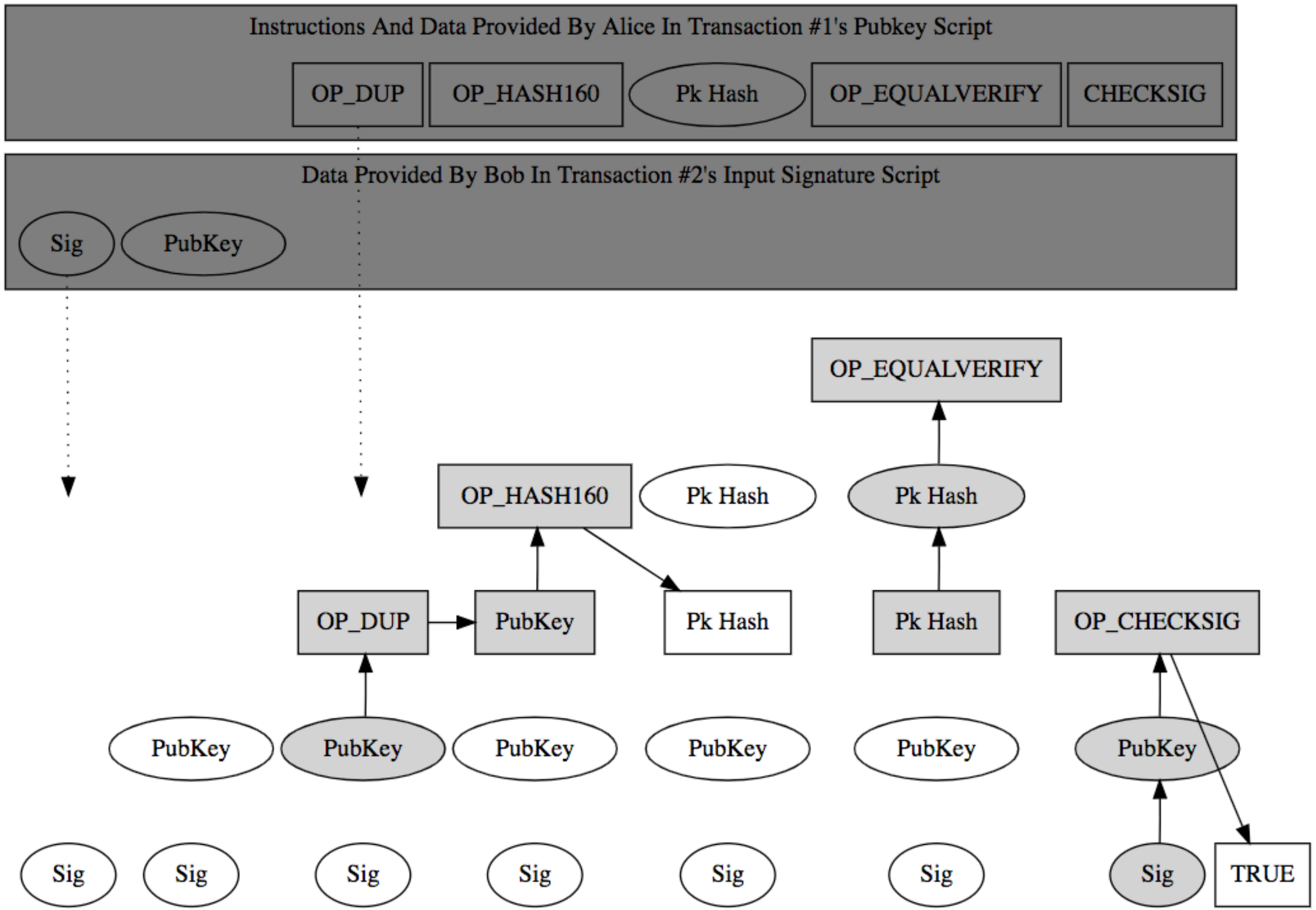
Spending A P2PKH Output

P2PKH

- Bob’s signature script
  - His public key (script can check its hash)
  - Signature (proves possession of the corresponding private key and makes the transaction secure)



P2PKH Script Validation



Evaluation Stack Over Time During Succesful P2PKH Script Validation

P2PKH Example

metadata

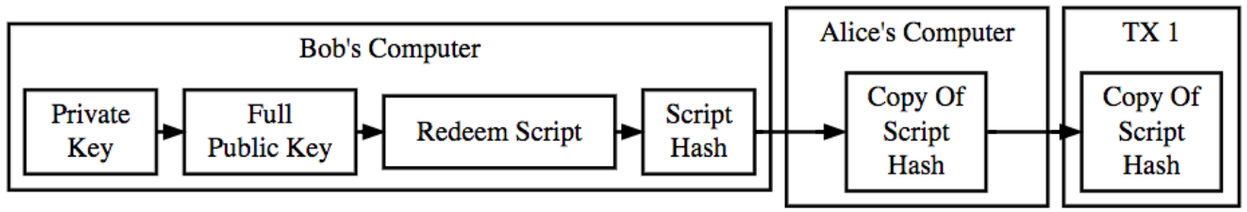
input(s)

output(s)

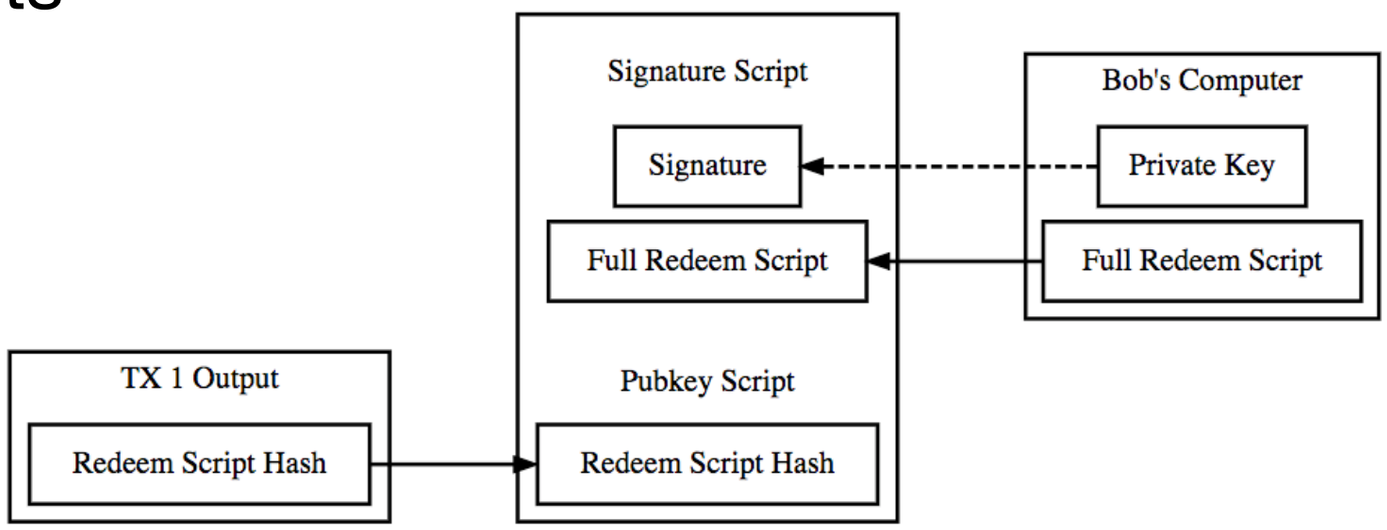

```
{
    "hash":"5a42590fbe0a90ee8e8747244d6c84f0db1a3a24e8f1b95b10c9e050990b8b6b",
    "ver":1,
    "vin_sz":2,
    "vout_sz":1,
    "lock_time":0,
    "size":404,
    "in":[
     {
      "prev_out":{
       "hash":"3be4ac9728a0823cf5e2deb2e86fc0bd2aa503a91d307b42ba76117d79280260",
       "n":0
      },
       "scriptSig":"30440....3f3a4ce81"
     },
     {
      "prev_out":{
       "hash":"7508e6ab259b4df0fd5147bab0c949d81473db4518f81afc5c3f52f91ff6b34e",
       "n":0
      },
      "scriptSig":"304602210....3f3a4ce81"
     }
    ],
    "out":[
     {
      "value":"10.12287097",
      "scriptPubKey":"OP_DUP OP_HASH160 69e02e18b5705a05dd6b28ed517716c894b3d42e OP_EQUALVERIFY OP_CHECKSIG"
     }
    ]
}
```

Other Scripts

- Pay-To-Script-Hash (P2SH)
  - (Almost) arbitrary spending rules
  - Possible to create contracts
    - Multisig (m-of-n)
    - Escrow
    - …
  - Only some of them are supported in practice



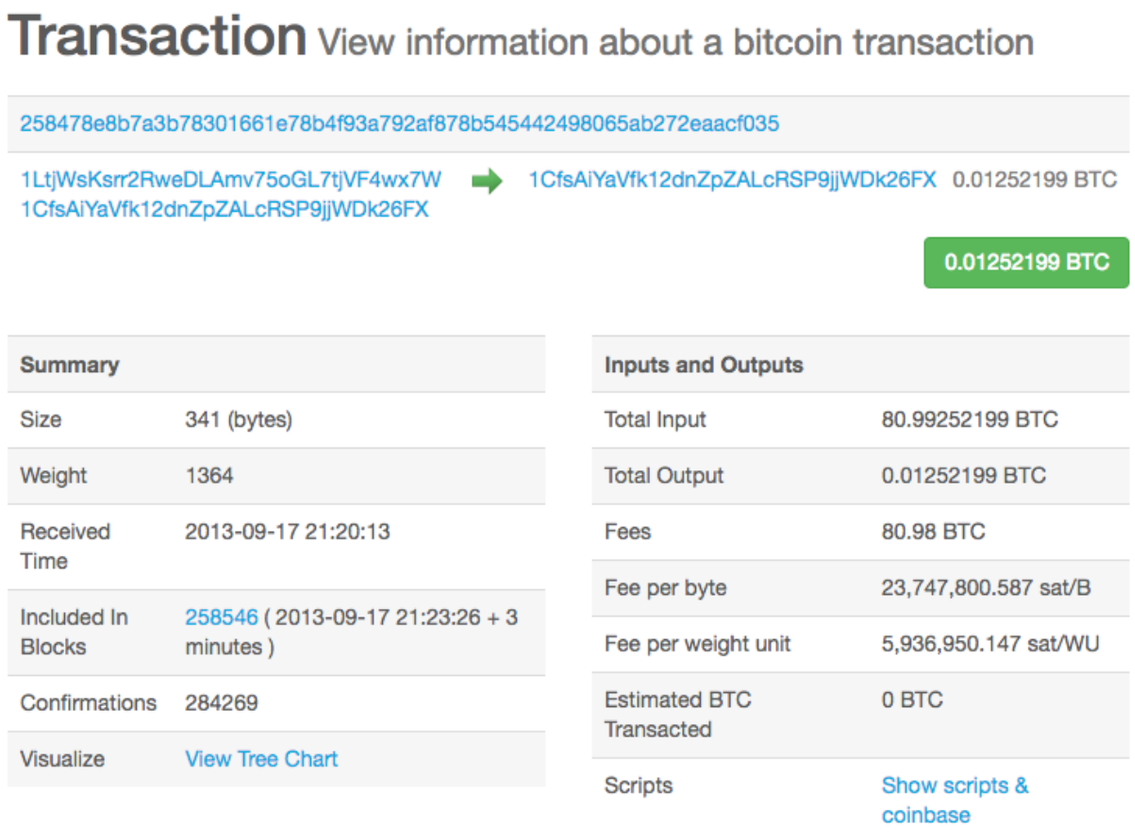
Creating A P2SH Redeem Script Hash To Receive Payment



Spending A P2SH Output

Transactions (Misc)

- OP_RETURN <data>
  - Returns <data>, cannot be spent
- Fees
  - Price per byte
  - Prioritize transactions
  - Change outputs
  - Mistaken transaction
    - No change output specified



Transaction View information about a bitcoin transaction

258478e8b7a3b78301661e78b4f93a792af878b545442498065ab272eaacf035

1LtjWsKsrr2RweDLAmv75oGL7tjVF4wx7W
1CfsAiYaVfk12dnZpZALcRSP9jjWDk26FX

1CfsAiYaVfk12dnZpZALcRSP9jjWDk26FX 0.01252199 BTC

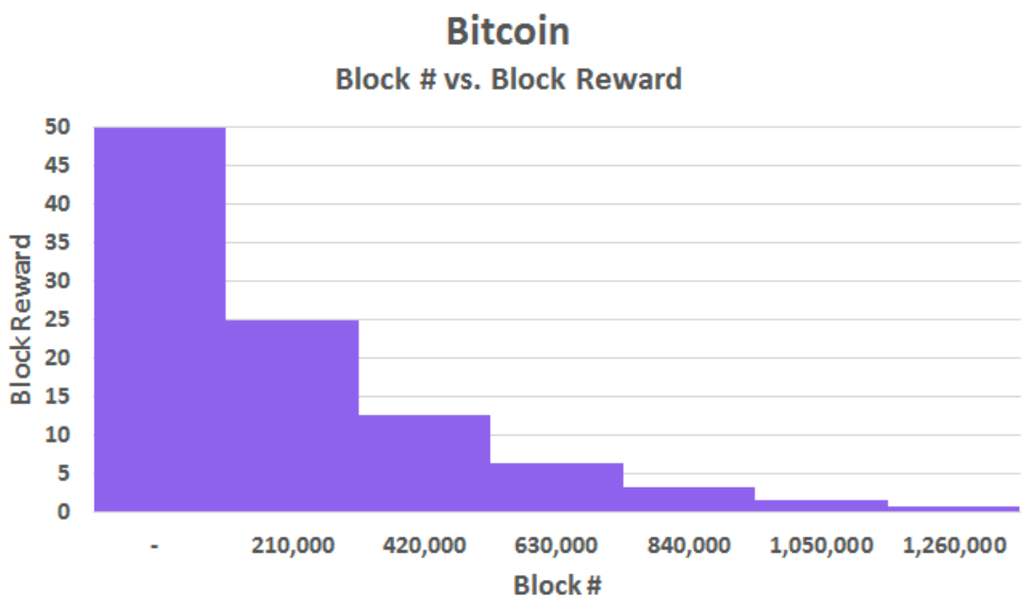
0.01252199 BTC

Summary	
Size	341 (bytes)
Weight	1364
Received Time	2013-09-17 21:20:13
Included In Blocks	258546 (2013-09-17 21:23:26 + 3 minutes)
Confirmations	284269
Visualize	View Tree Chart

Inputs and Outputs	
Total Input	80.99252199 BTC
Total Output	0.01252199 BTC
Fees	80.98 BTC
Fee per byte	23,747,800.587 sat/B
Fee per weight unit	5,936,950.147 sat/WU
Estimated BTC Transacted	0 BTC
Scripts	Show scripts & coinbase

Coin Minting

- But how coins are actually introduced to the system?
- **Coinbase transactions** (the first transaction of every block)
  - Collecting and spending block reward plus transaction fees paid by transactions of this block
    - Coinbase coins cannot be spent for at least 100 blocks
      - Prevent immediate spending blocks that may turn out to be stale
- Reward scheme



Genesis Block

- Input field of a coinbase transaction can be an arbitrary text

```
00000000   01 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
00000010   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
00000020   00 00 00 00 3B A3 ED FD   7A 7B 12 B2 7A C7 2C 3E   ....;£íýz{.²zÇ,>
00000030   67 76 8F 61 7F C8 1B C3   88 8A 51 32 3A 9F B8 AA   gv.a.È.Ã^ŠQ2:Ÿ¸ª
00000040   4B 1E 5E 4A 29 AB 5F 49   FF FF 00 1D 1D AC 2B 7C   K.^J)«_Iÿÿ...¬+|
00000050   01 01 00 00 00 01 00 00   00 00 00 00 00 00 00 00   ................
00000060   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
00000070   00 00 00 00 00 00 FF FF   FF FF 4D 04 FF FF 00 1D   ......ÿÿÿÿM.ÿÿ..
00000080   01 04 45 54 68 65 20 54   69 6D 65 73 20 30 33 2F   ..EThe Times 03/
00000090   4A 61 6E 2F 32 30 30 39   20 43 68 61 6E 63 65 6C   Jan/2009 Chancel
000000A0   6C 6F 72 20 6F 6E 20 62   72 69 6E 6B 20 6F 66 20   lor on brink of
000000B0   73 65 63 6F 6E 64 20 62   61 69 6C 6F 75 74 20 66   second bailout f
000000C0   6F 72 20 62 61 6E 6B 73   FF FF FF FF 01 00 F2 05   or banksÿÿÿÿ..ò.
000000D0   2A 01 00 00 00 43 41 04   67 8A FD B0 FE 55 48 27   *....CA.gŠý°þUH'
000000E0   19 67 F1 A6 71 30 B7 10   5C D6 A8 28 E0 39 09 A6   .gñ¦q0·.\Ö¨(à9.¦
000000F0   79 62 E0 EA 1F 61 DE B6   49 F6 BC 3F 4C EF 38 C4   ybàê.aÞ¶Iö¼?Lï8Ä
00000100   F3 55 04 E5 1E C1 12 DE   5C 38 4D F7 BA 0B 8D 57   óU.å.Á.Þ\8M÷º..W
00000110   8A 4C 70 2B 6B F1 1D 5F   AC 00 00 00 00            ŠLp+kñ._¬....
```


THE TIMES

Saturday January 3 2009 timesonline.co.uk No 69523 £1.50

Eat Out from £5

More than 900 great restaurants, including four Gordon Ramsay favourites from £15

Start collecting tokens today · Pullout inside

Israel prepares to send tanks and troops into Gaza

Israel allowed foreigners to flee the Gaza Strip as it prepared for a ground offensive. At least 430 Palestinians were killed in a week of airstrikes News, page 3

Chancellor on brink of second bailout for banks

Billions may be needed as lending squeeze tightens

Francis Elliott Deputy Political Editor
Gary Duncan Economics Editor

Alistair Darling has been forced to consider a second bailout for banks as the lending drought worsens.

The Chancellor will decide within weeks whether to pump billions more into the economy as evidence mounts that the £37billion part-nationalisation last year has failed to keep credit flowing. Options include cash injections, offering banks cheaper state guarantees to raise money privately or buying up "toxic assets", The Times has learnt.

The Bank of England revealed yesterday that, despite intense pressure, the banks curbed lending in the final quarter of last year and plan even tighter restrictions in the coming months. Its findings will alarm the Treasury.

The Bank is expected to take yet more aggressive action this week by cutting the base rate from its current level of 2 per cent. Doing so would reduce the cost of borrowing but have little effect on the availability of loans.

Whitehall sources said that ministers planned to "keep the banks on the hook" but accepted that they need more help to restore lending levels. Formally, the Treasury plans to focus on state-backed guarantees to encourage private finance, but a number of interventions are on the table, including further injections of taxpayers' cash.

Under one option, a "bad bank" would be created to dispose of bad debts. The Treasury would take bad loans off the hands of troubled banks, perhaps swapping them for government bonds. The toxic assets, blamed for poisoning the financial system, would be parked in a state vehicle or "bad bank" that would manage them and attempt to dispose of them while "detoxifying" the mainstream banking system.

The idea would mirror the initial proposal by Henry Paulson, the US Treasury Secretary, to underpin the American banking system by buying

99p

Pub chain cuts the price of a pint from £1.49 to 1989 levels Business, page 67

Continued on page 6, col 5
Leading article, page 2

Michael Sheen Frost, Nixon and me Magazine

Working mums So that's how she does it Body&Soul

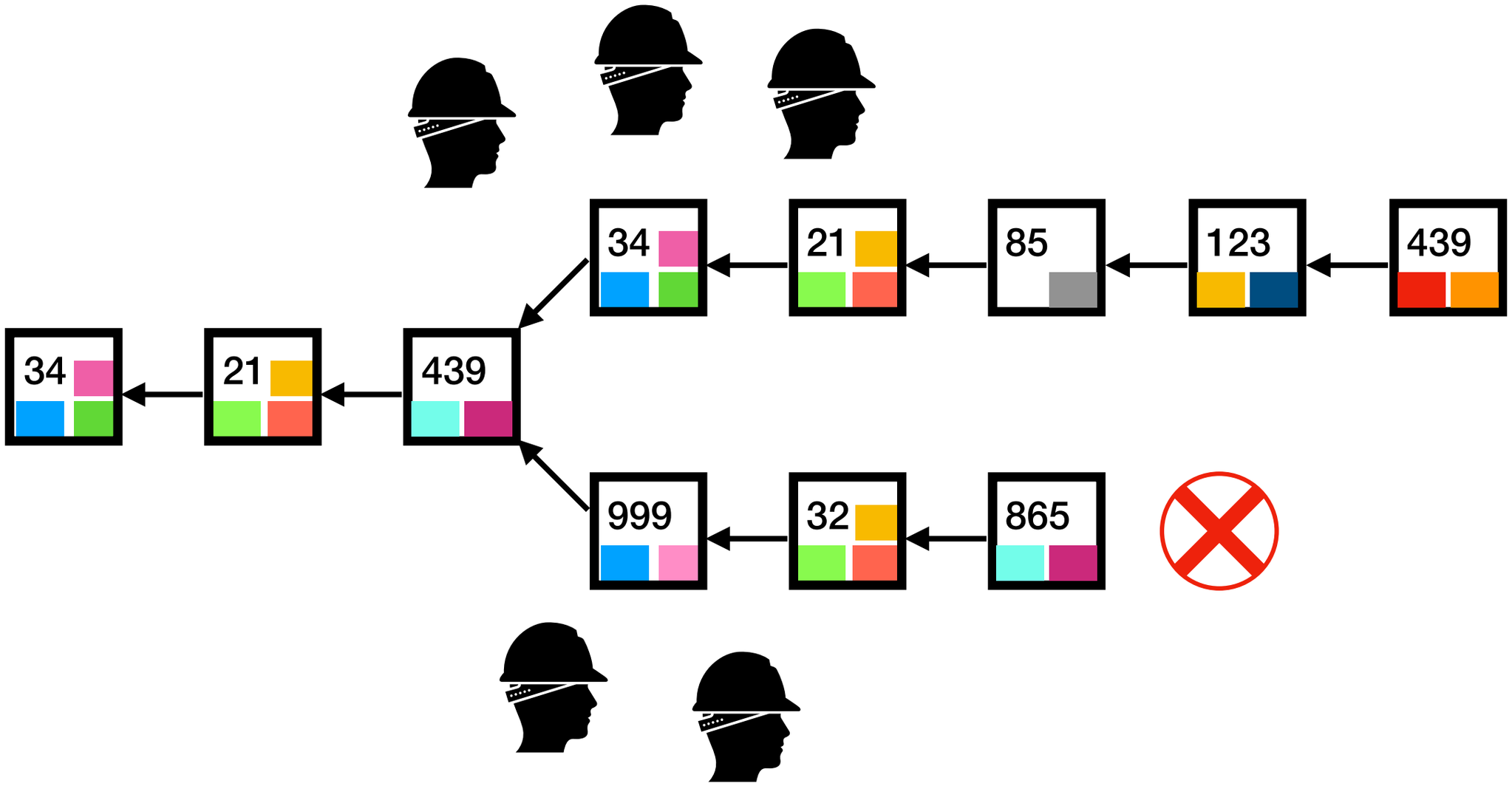
Detox in style The best spas on the planet Travel

Salmon Rushdie I Won't Marry Again Pages 23, 25

Giant Killing? Guide to the FA Cup Third Round Sport

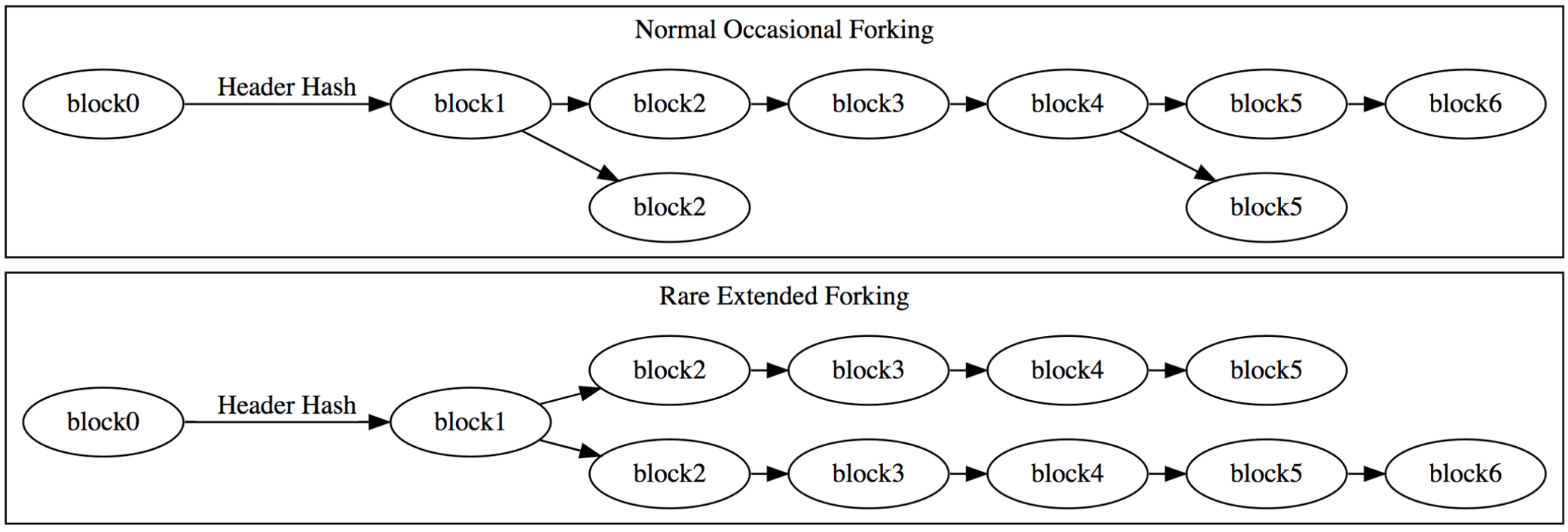
Forks

Forks (Recap)

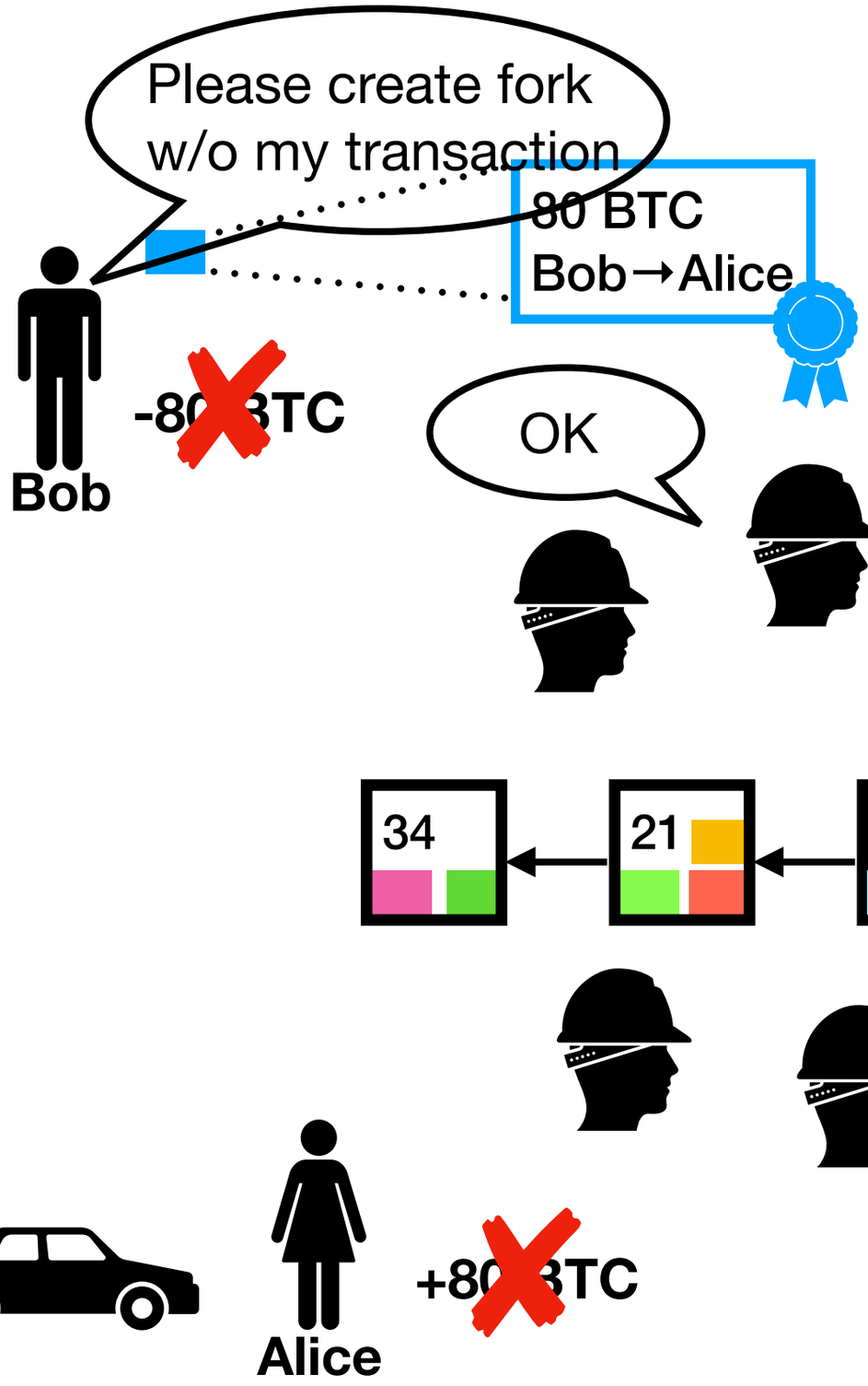


Forks

- In the case of one-block fork, nodes individually decide which to accept (usually, the first seen)
- Nodes follow the strongest chain and discard stale blocks of weaker chains
    - Sometimes these stale blocks are (incorrectly) called orphans
- Why forks are bad?



Double Spending

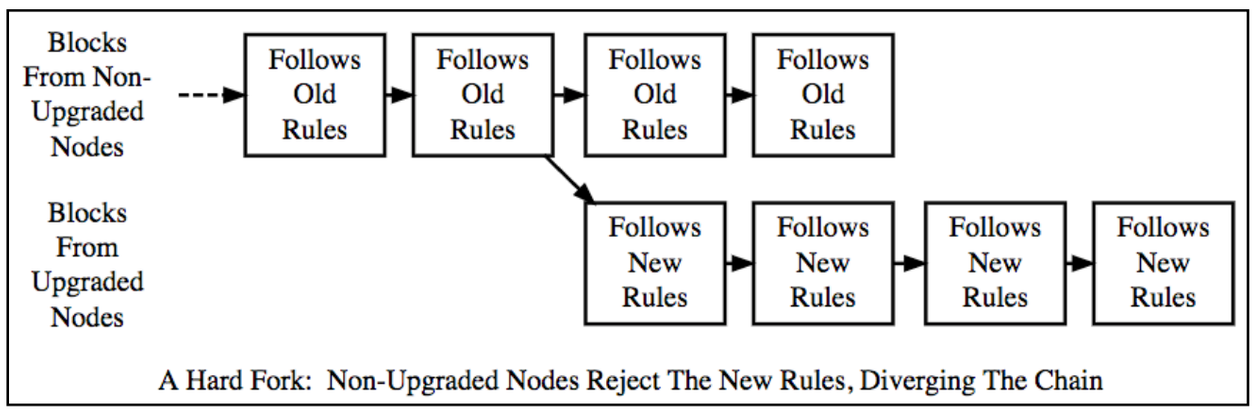


Double Spending Protection

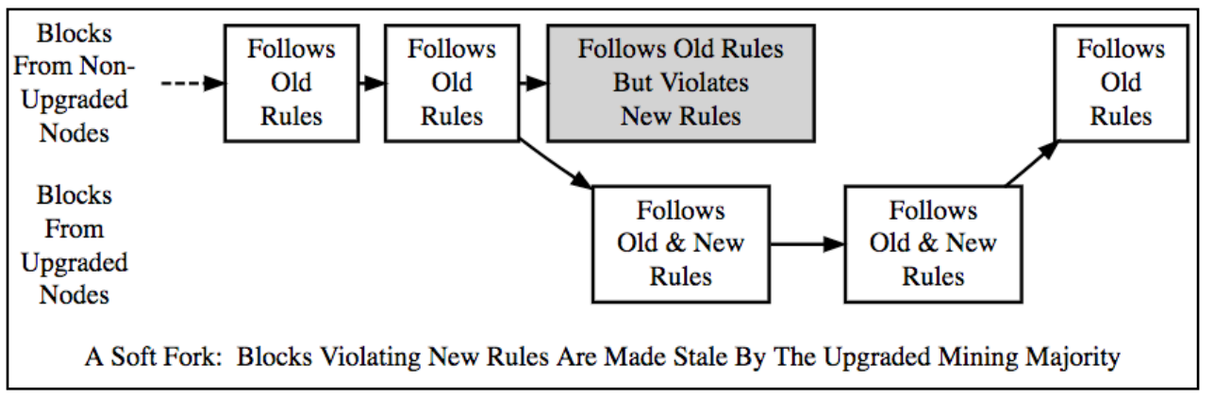
- Confirmations
  - With honest majority of hash power long forks are unlikely
    - 6 confirmation blocks suggested (~1h)
- Software protection
  - Checkpoints implemented
    - (actually, that is a protocol violation)

Consensus Rule Changes

- Nodes use the same consensus rules (replicated state machine), so what about updates (new features, bug fixes)?
- Updated and non-updated nodes will follow different rules
- Hard fork: block following the new consensus rules is accepted by upgraded but rejected by non-upgraded nodes
    - Chains will be divergent, nodes will never agree
- Soft fork: block violating the new consensus rules is rejected by upgraded but accepted by non-upgraded nodes
    - Chains will converge, nodes will agree (when majority of the hash rate applies new rules)
- Examples: changing block size, rejecting malformed transactions, …
- Waiting for hash power, flag days, user/miner activated soft forks



A Hard Fork: Non-Upgraded Nodes Reject The New Rules, Diverging The Chain



A Soft Fork: Blocks Violating New Rules Are Made Stale By The Upgraded Mining Majority

Reading

- Textbook 2,3,4
- https://bitcoin.org/en/developer-guide
- https://queue.acm.org/detail.cfm?id=3136559