COMP4137 Blockchain Technology and Applications
COMP7200 Blockchain Technology

Lecturer: Dr. Hong-Ning Dai (Henry)

Lecture 5

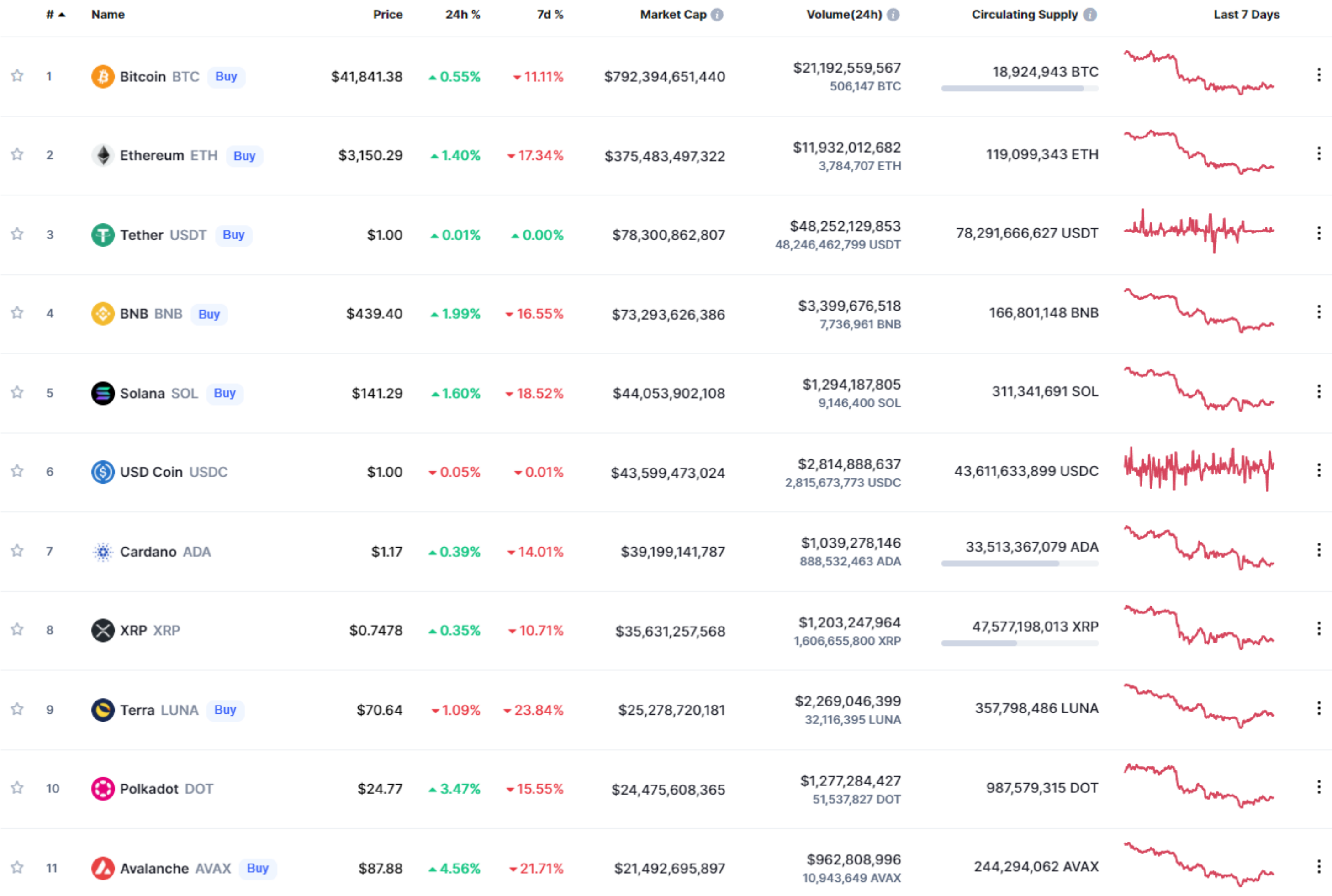
Bitcoin Details

Outline

- Bitcoin Block Format
  - Header
  - Body
- Bitcoin Consensus
  - Mining
  - Target Threshold
- Bitcoin Transactions
  - Transaction Format
  - Script

Cryptocurrency

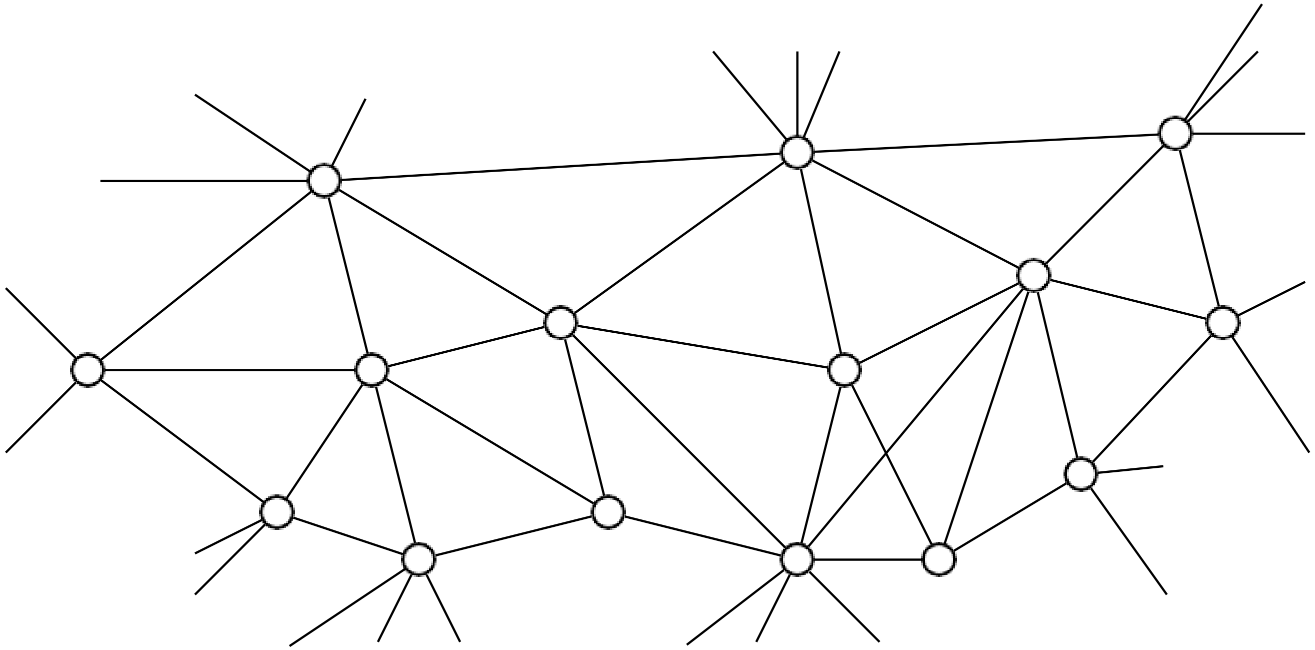
- Over 2000 cryptocurrencies at the moment



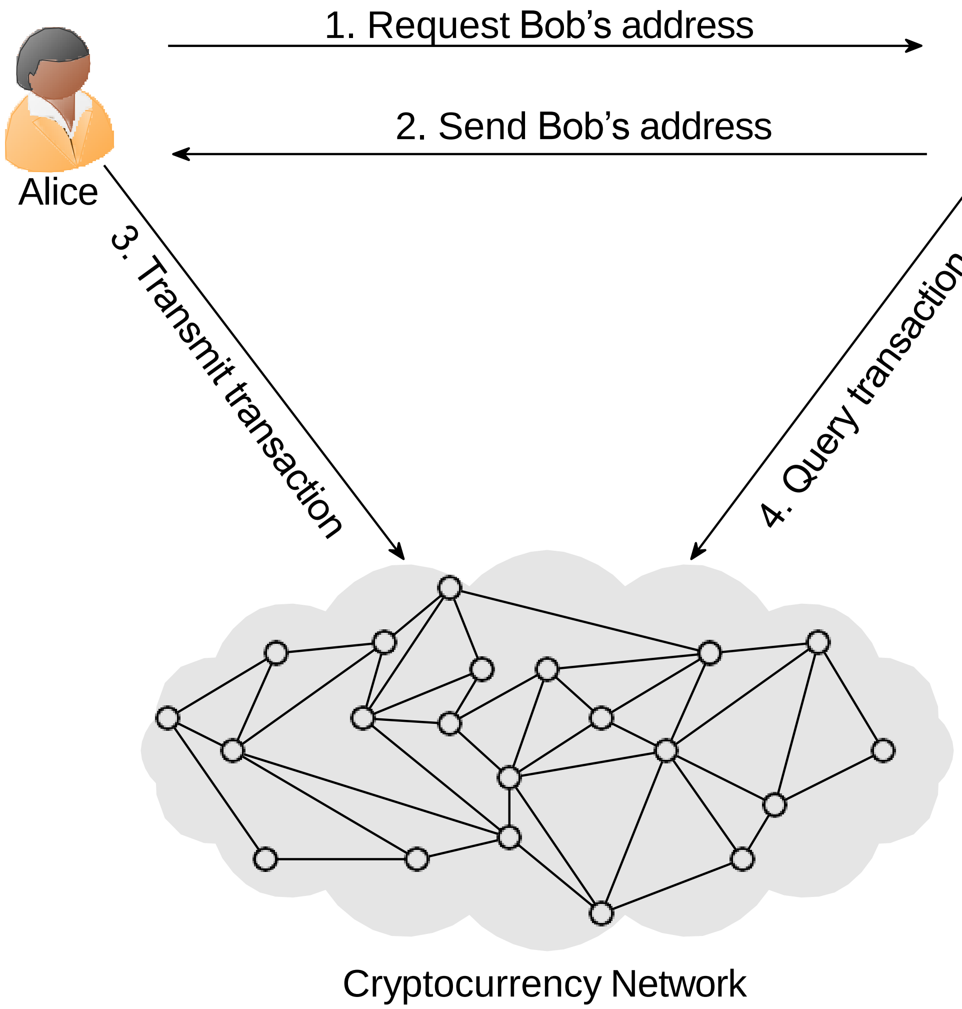
# ▲	Name	Price	24h %	7d %	Market Cap	Volume(24h)	Circulating Supply	Last 7 Days
1	Bitcoin BTC Buy	$41,841.38	▲0.55%	▼11.11%	$792,394,651,440	$21,192,559,567 506,147 BTC	18,924,943 BTC	
2	Ethereum ETH Buy	$3,150.29	▲1.40%	▼17.34%	$375,483,497,322	$11,932,012,682 3,784,707 ETH	119,099,343 ETH	
3	Tether USDT Buy	$1.00	▲0.01%	▲0.00%	$78,300,862,807	$48,252,129,853 48,246,462,799 USDT	78,291,666,627 USDT	
4	BNB BNB Buy	$439.40	▲1.99%	▼16.55%	$73,293,626,386	$3,399,676,518 7,736,961 BNB	166,801,148 BNB	
5	Solana SOL Buy	$141.29	▲1.60%	▼18.52%	$44,053,902,108	$1,294,187,805 9,146,400 SOL	311,341,691 SOL	
6	USD Coin USDC	$1.00	▼0.05%	▼0.01%	$43,599,473,024	$2,814,888,637 2,815,673,773 USDC	43,611,633,899 USDC	
7	Cardano ADA	$1.17	▲0.39%	▼14.01%	$39,199,141,787	$1,039,278,146 888,532,463 ADA	33,513,367,079 ADA	
8	XRP XRP	$0.7478	▲0.35%	▼10.71%	$35,631,257,568	$1,203,247,964 1,606,655,800 XRP	47,577,198,013 XRP	
9	Terra LUNA Buy	$70.64	▼1.09%	▼23.84%	$25,278,720,181	$2,269,046,399 32,116,395 LUNA	357,798,486 LUNA	
10	Polkadot DOT	$24.77	▲3.47%	▼15.55%	$24,475,608,365	$1,277,284,427 51,537,827 DOT	987,579,315 DOT	
11	Avalanche AVAX Buy	$87.88	▲4.56%	▼21.71%	$21,492,695,897	$962,808,996 10,943,649 AVAX	244,294,062 AVAX	

What is Bitcoin?

- Cryptocurrency
- Open source
- Decentralized network

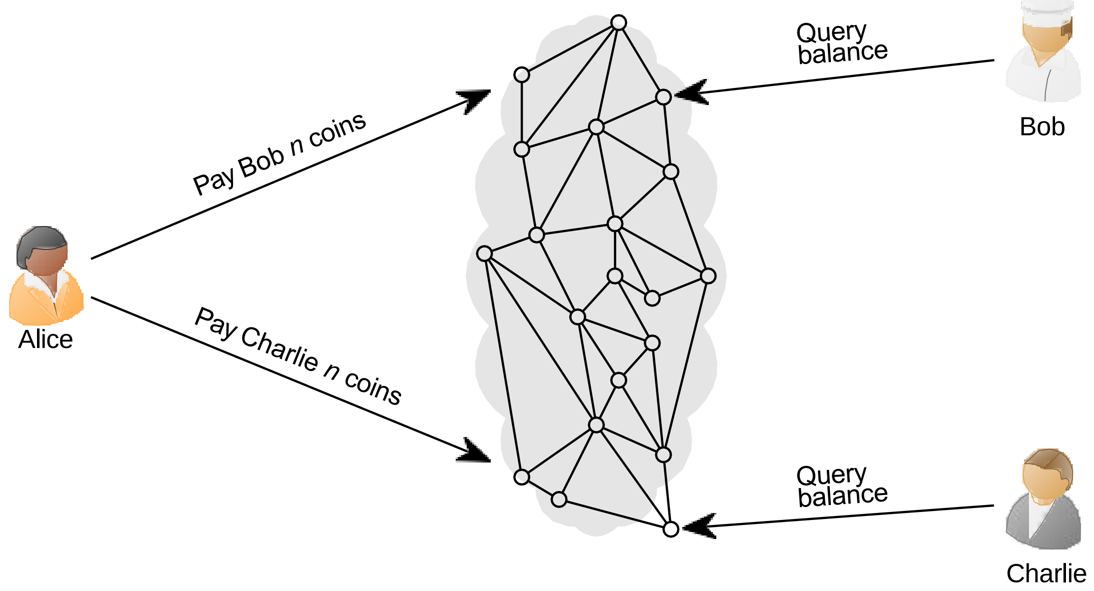


Cryptocurrency Transaction Workflow



Decentralization Challenges

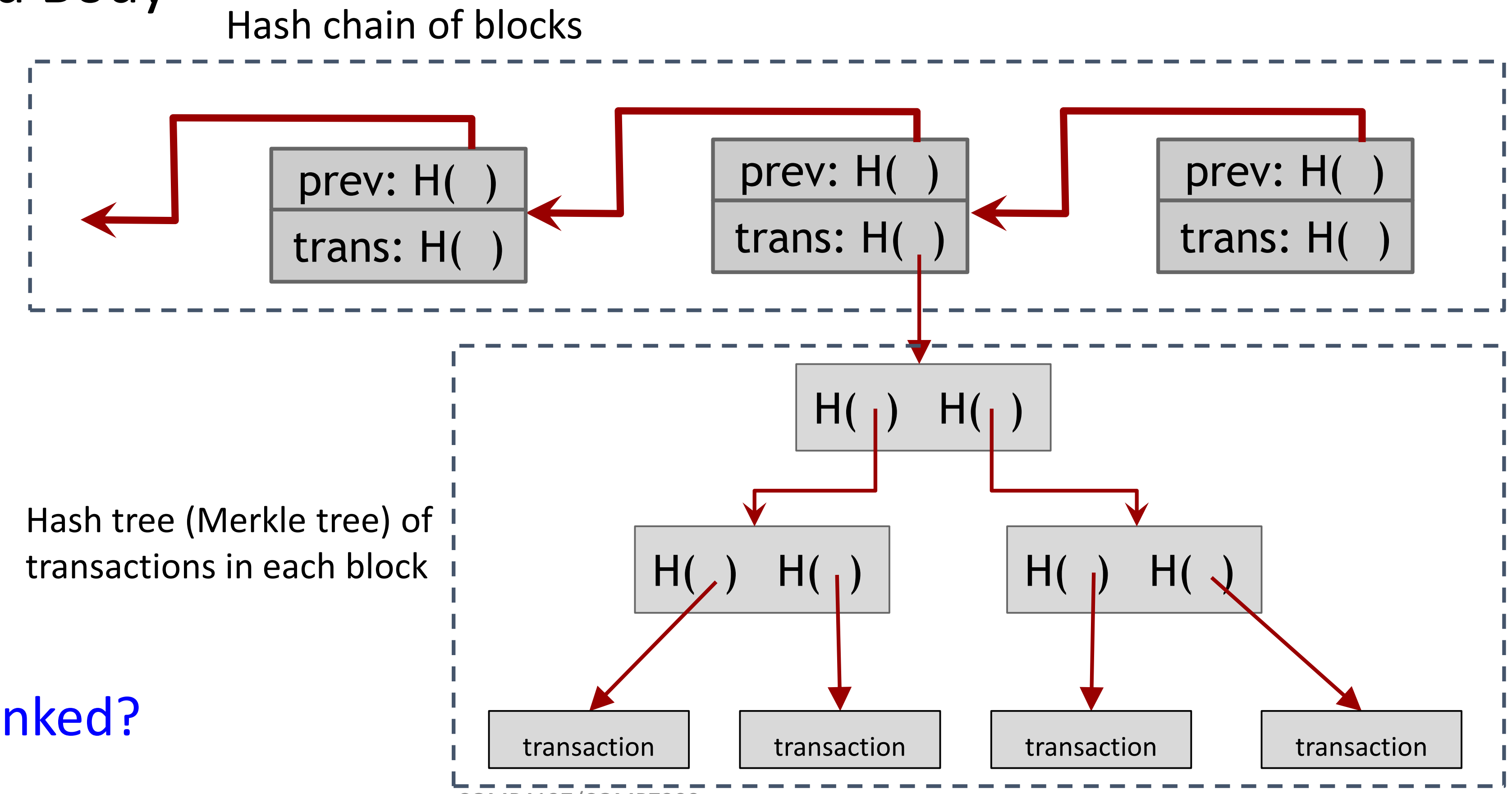
- Counterfeiting
- Currency creation rules
- Double spending
  - Alice pays Bob *n* digital coins for a cake
  - Alice uses the **same** *n* digital coins to pay Charlie for a book



Solution without a central coordinator?

The Blockchain

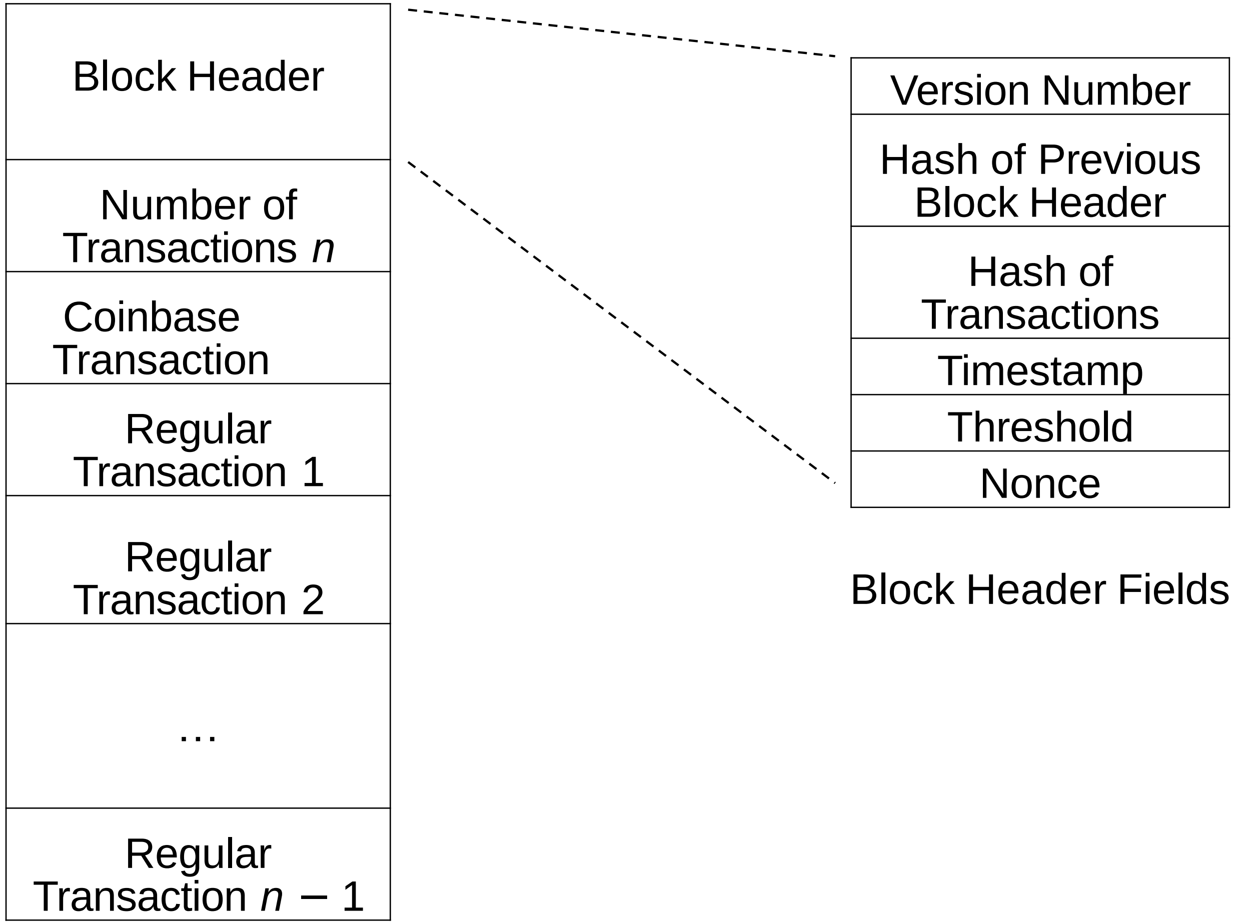
- **Blockchain**: A public ledger (database) to store all transactions which is replicated by many network nodes
  - Header and Body



Hash tree (Merkle tree) of transactions in each block

How are blocks linked?

Bitcoin Block Format



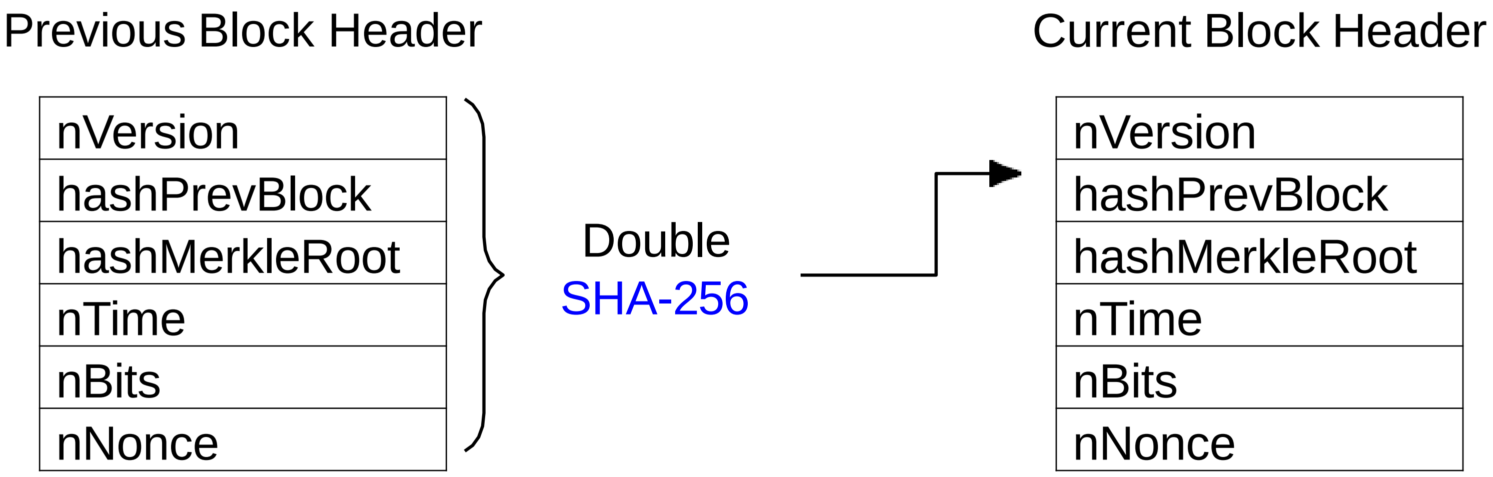
Block Header Fields

- Hash = Output of cryptographic hash function

Block Header

nVersion	4 bytes
hashPrevBlock	32 bytes
hashMerkleRoot	32 bytes
nTime	4 bytes
nBits	4 bytes
nNonce	4 bytes

80 bytes



SHA256(SHA256(header))

Cryptographic Hash Functions (CHF)

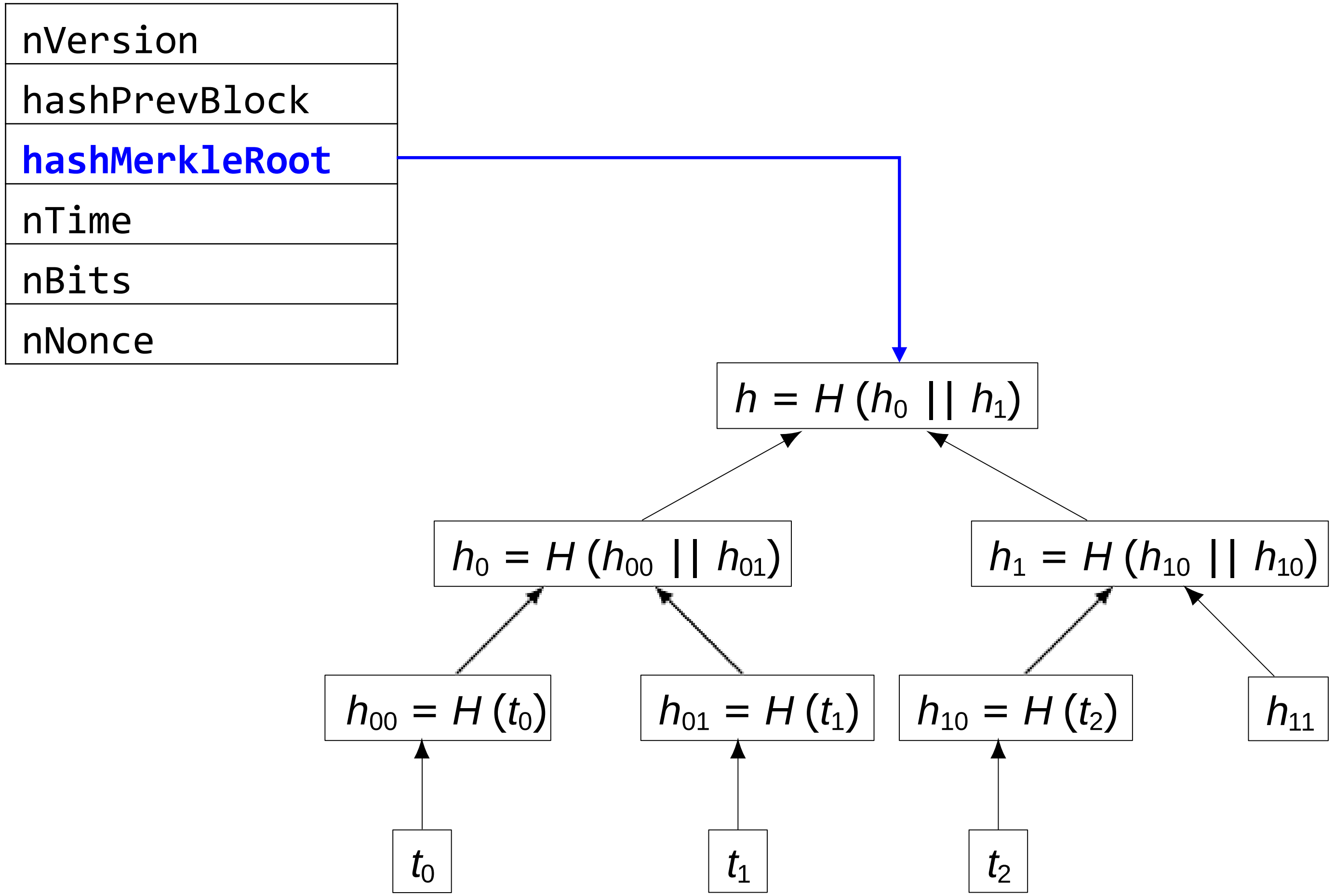
- Easy to compute but difficult to invert
- Collision-resistant
- Pseudorandom outputs
- SHA-256 = NIST approved CHF with 256-bit outputs

Input	SHA-256 Output
july0	171c9f5053d5d675d1d1ed477c908e98498e6751ae392a78807c3cd6ad6975fa
july1	7d8033d140d8b8db8324753a25c5e32ee4faa9c4e306bddb317907be51cd8a24
july2	bda0b2ab2c7d654589b32f46a548cba27b7371f27b070ddd7d3b87122a078f06
july3	dfa3569a46b1a13c24c9f385da140f4763a3fbb70f8eebe0f29ba535145d32ca
july4	27d39d26edc54c11cc78d17bf0dd294413300dd004127fa6dcff368ea74bb87c
july5	a0ebd3e23823fc291b090abd2eb1403912be6b72398f3bf4e92c4ec555902d53
july6	dc7d6bcc266af402e53b9fb978b6579940bb97743f6e975a988cb20d903e0c5f
july7	984906fbbaa7dbad2ee01a81df7a237bfdb63aeb06b4cf97a89fc004542c1dab
july8	7be4d491b73a4797304980070d5b5fb5c7fd6921e70efc7ce38023c50664803d
july9	e8c4af8895bcddb9cea3e3e1e8a08e090690bb55fd6617da5aa0873f27e218ee

- Hex digits: 0 = 0000,1 = 0001,2 = 0010,..., a = 1010, b = 1011, c = 1100,..., e = 1110, f = 1111
- At a billion outputs per second, 78 billion years required to calculate 2^{100} outputs

Merkle Root in Block Header

- `hashMerkleRoot` contains root hash of transaction Merkle tree
- Modifying any transaction will modify the block header



nVersion

hashPrevBlock

hashMerkleRoot

nTime

nBits

nNonce

$h = H(h_0 \mid\mid h_1)$

$h_0 = H(h_{00} \mid\mid h_{01})$

$h_1 = H(h_{10} \mid\mid h_{10})$

$h_{00} = H(t_0)$

$h_{01} = H(t_1)$

$h_{10} = H(t_2)$

h_{11}

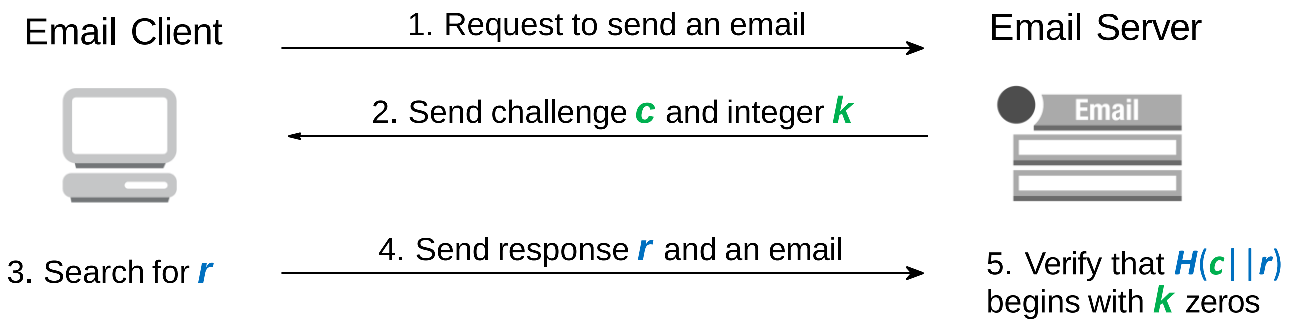
t_0

t_1

t_2

Hashcash

- A database you own where anyone in the world can add entries?
  - Your email inbox
- Hashcash was proposed in 1997 to prevent spam
- Protocol
  1. Suppose an email client wants to send email to an email server
     Client and server agree upon a cryptographic hash function H
  2. Server sends the client a challenge string c and an integer k
  3. Client needs to find a string r s.t. $H(c||r)$ begins with k zeros



Email Client

Email Server

1. Request to send an email

2. Send challenge c and integer k

3. Search for r

4. Send response r and an email

5. Verify that $H(c||r)$ begins with k zeros

- The r is considered ***proof-of-work (PoW)***
  - Difficult to generate but easy to verify

Hashcash Proof of Work

- Public Challenge: $\boldsymbol{c}$

SHA256

- Goal: Find nonce $\boldsymbol{r}$ s.t. $\boldsymbol{H}(\boldsymbol{c}||\boldsymbol{r}) = \underbrace{00\cdots00}_{k}1\cdots\cdots\cdots$

- The probability to find such nonce

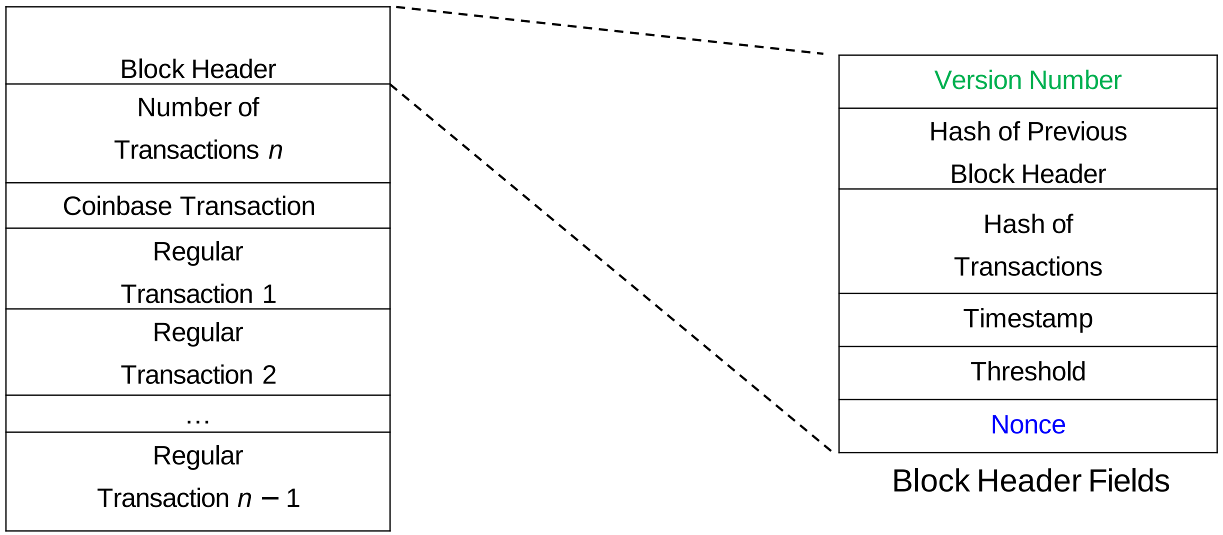
$$\Pr[\text{first } k \text{ bits of } \boldsymbol{H}(\boldsymbol{c}||\boldsymbol{r}) \text{ are zeros}] = \frac{1}{2^k}$$

Outline

- Bitcoin Block Format
    - Header
    - Body
- Bitcoin Consensus
    - Mining
    - Target Threshold
- Bitcoin Transactions
    - Transaction Format
    - Script

Bitcoin Mining

- Mining = Process of adding new blocks to the blockchain
- Nodes perform transactions and broadcast them
- Miners collect some of these transactions into a candidate block



Block Header Fields

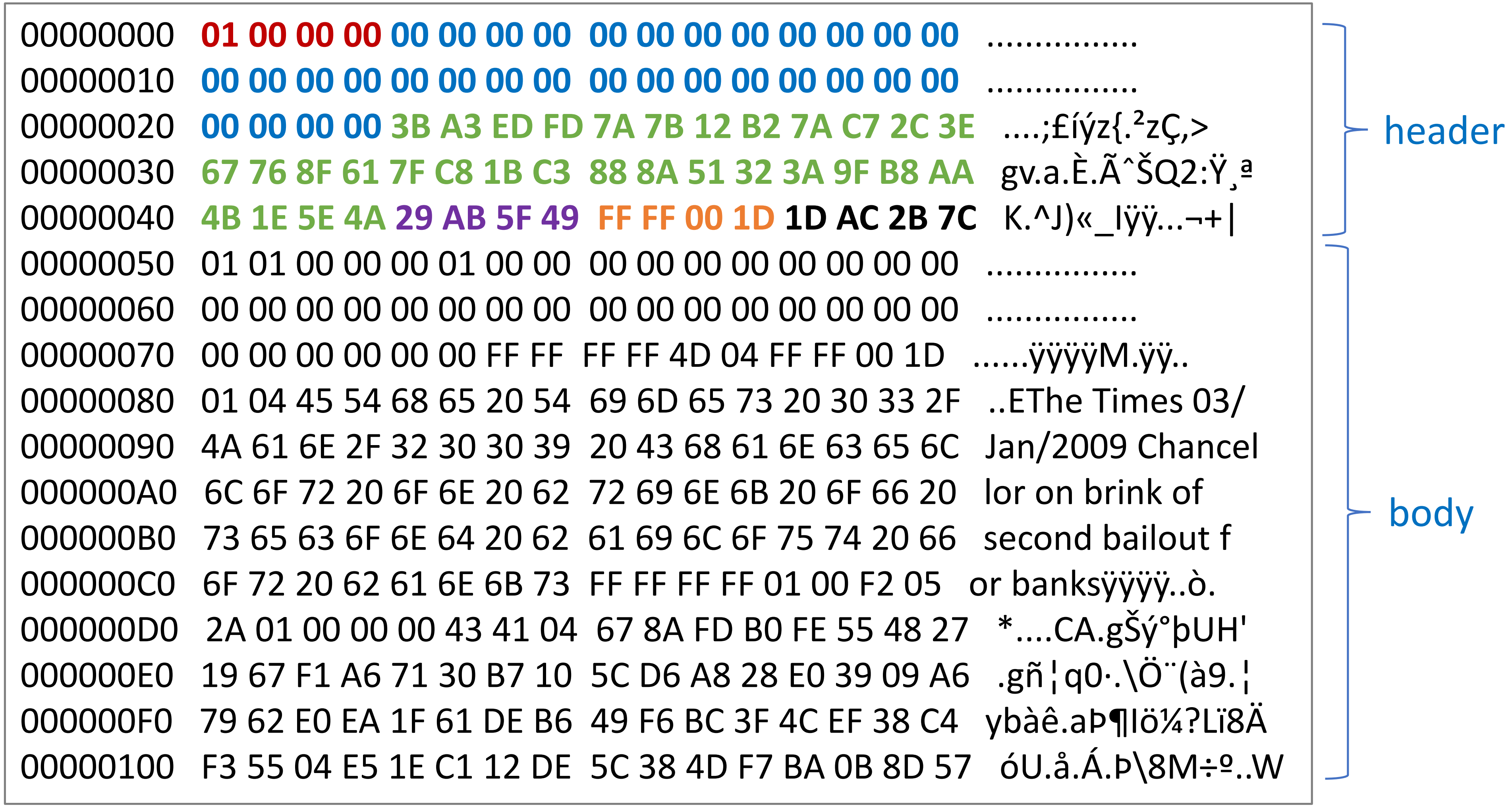
- Threshold encodes a 256-bit value like $0\mathrm{x}\underbrace{00\cdots00}_{16}\underbrace{\mathrm{FFFF}\ \ldots\ \mathrm{FFFFF}}_{48}$
- Miner who can find Nonce such that $\mathrm{SHA256}(\mathrm{SHA256}(\underbrace{\mathrm{VersionNumer} \parallel \cdots \parallel \mathrm{Nonce}}_{\text{Candidate Block Header}})) \leq \mathrm{Threshold}$ can add a new block.

Mining is Hard

Target value T	Fraction of Double SHA256's output $\leq T$	
$0\text{x}7\,\underbrace{\text{FFFF} \ \ldots \ \text{FFFFF}}_{63}$	$\frac{1}{2}$	$\frac{[0 \ldots 7] \ \to 8}{16}$
$0\text{x}0\,\underbrace{\text{FFFF} \ \ldots \ \text{FFFFF}}_{63}$	$\frac{1}{16}$	$\frac{1}{2^4}$
$0\text{x}\,\underbrace{00 \cdots 00}_{16}\,\underbrace{\text{FFFF} \ \ldots \ \text{FFFFF}}_{48}$	$\frac{1}{2^{64}}$	$\frac{1}{2^{4*16}}$

$$\Pr[\text{DoubleSHA256's output} \leq T)] \approx \frac{T+1}{2^{256}}$$

Genesis Block (Raw Hex Version)



```
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
```

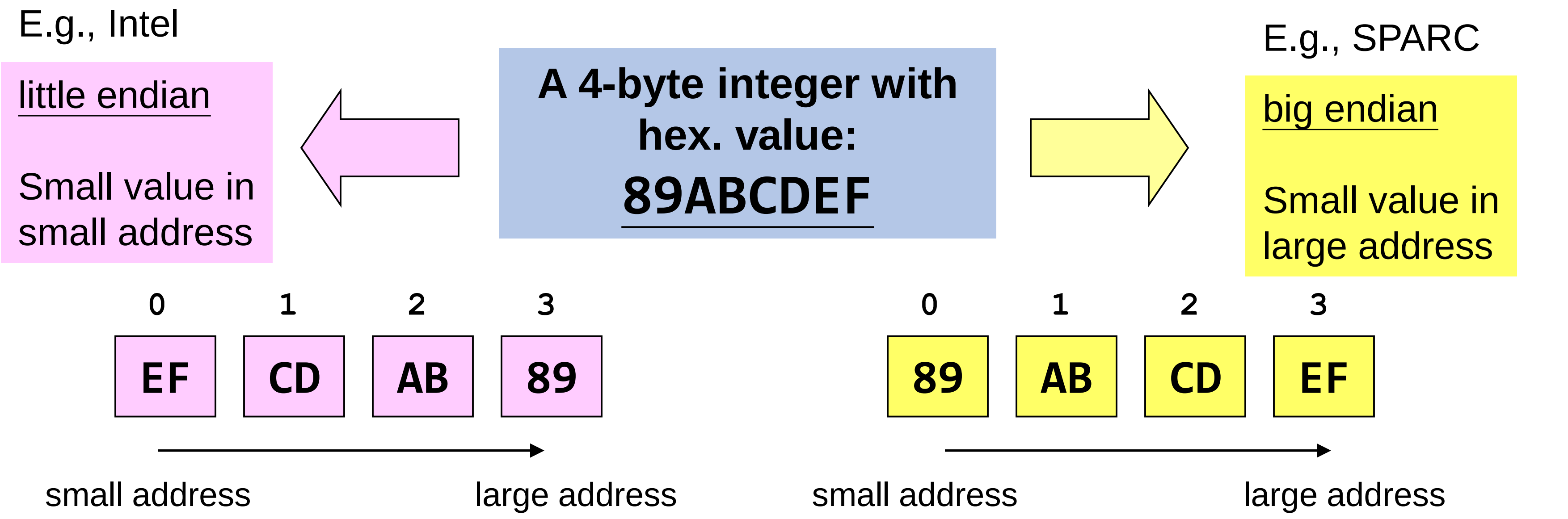
Explanation (in the next page)

Genesis Block Header Explained

- ***nVersion***: 01 00 00 00 (= 00 00 00 01, little endian)
- ***hashPrevBlock***:
  00000000000000000000000000000000000000000
  00000000000000000000000 (why?)
- ***hashMerkleRoot***:
  3BA3EDFD7A7B12B27AC72C3E67768F617FC81B
  C3888A51323A9FB8AA4B1E5E4A
- ***nTime***: 29 AB 5F 49
  - (= 49 5F AB 29 = 1231006505_{10}, little endian)
- ***nBits***: FF FF 00 1D
- ***nNonce***: 1D AC 2B 7C (= 2083236893_{10}, little endian)

Big Endian vs Little Endian

- Endian-ness is about byte ordering.
  - It means the way that a machine (we mean the entire computer architecture) orders the bytes.



E.g., Intel

little endian

Small value in small address

A 4-byte integer with hex. value:

89ABCDEF

E.g., SPARC

big endian

Small value in large address

0	1	2	3
EF	CD	AB	89

small address → large address

0	1	2	3
89	AB	CD	EF

small address → large address

Miner’s Incentive

- Block Reward
  1. Block Subsidy: each block contains a **coinbase transaction**, which creates 6.25 BTC
     - Each miner specifies his own address as the destination of the new coins
     - Every miner is competing to solve their own PoW puzzle

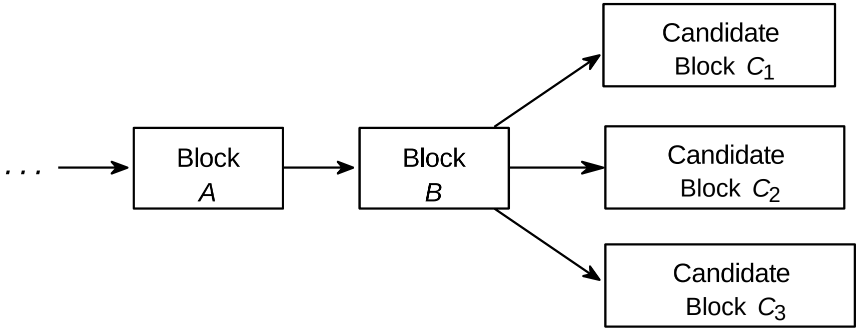
  2. Transaction fee: miners also collect transaction fees in the block

Mining Farms

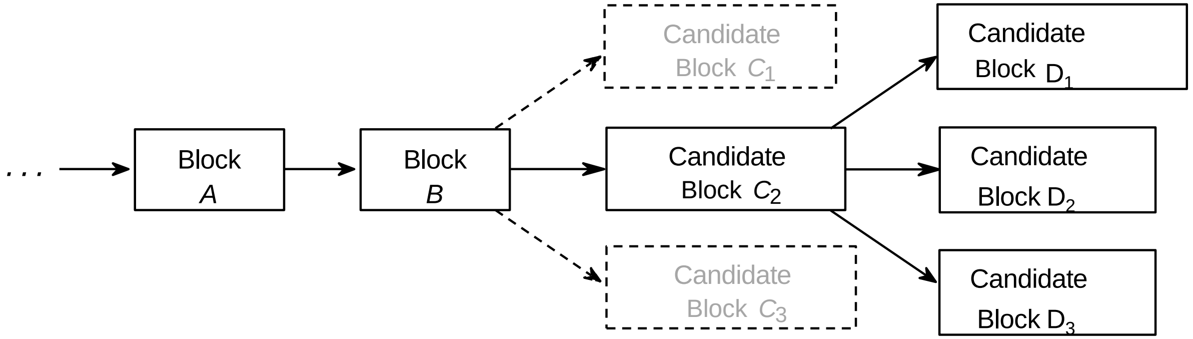
- Mining farms have thousands of mining rigs
- Each mining rig has dozens of mining chips
- Each chip has dozens of SHA256 mining cores
- Farms are located in places with cheap power and cooling

Block Addition Workflow

- Nodes broadcast transactions
- Miners accept valid transactions and reject invalid ones (solves double spending)
- Miners try to extend the latest block

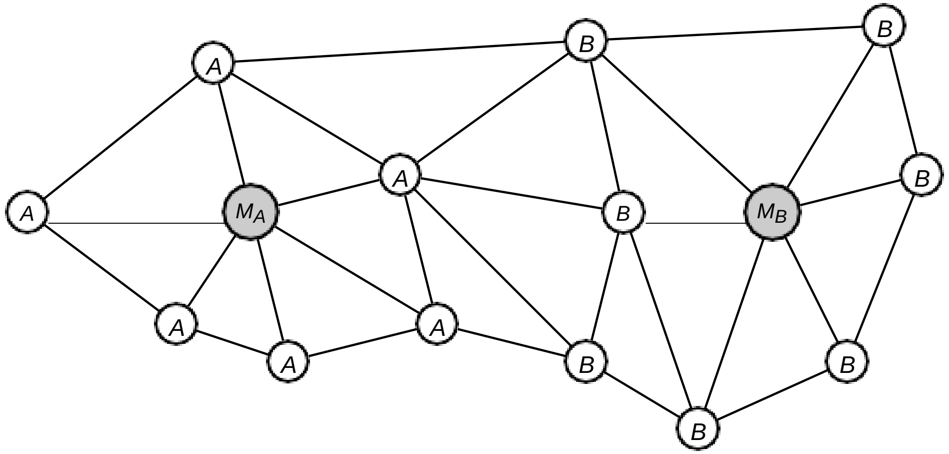


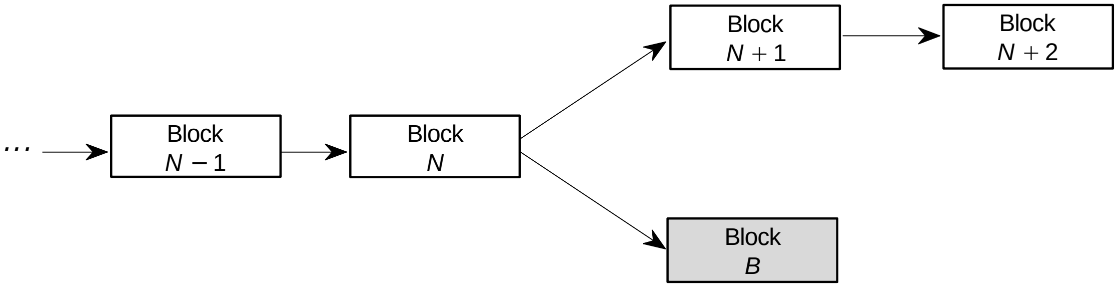
- Miners compete to solve the search puzzle and broadcast solutions
- Unsuccessful miners abandon their current candidate blocks and start work on new ones



What if two miners solve the puzzle at the same time?

- Both miners will broadcast their solutions on the network
- Nodes will accept the first solution they hear and reject others





$\cdots$

Block $N-1$

Block N

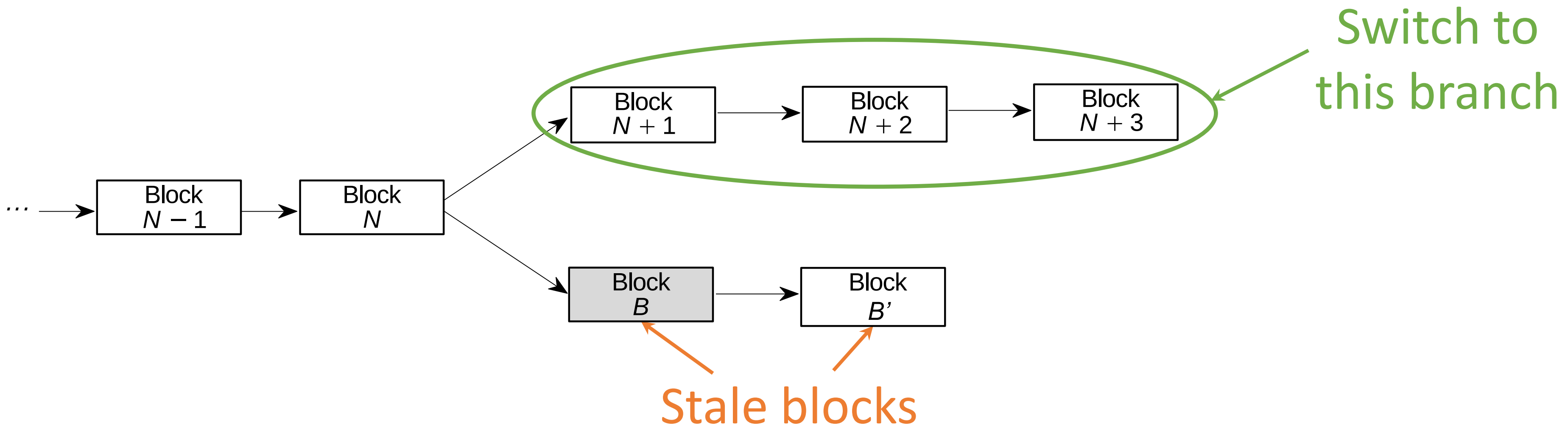
Block $N+1$

Block $N+2$

Block B

Blockchain fork

What if two miners solve the puzzle at the same time?



- Nodes always switch to the ***longest branch*** they become aware of
- Eventually the network will converge and achieve consensus
- This is called ***proof-of-work (PoW) consensus***

How often are new blocks created?

- Once every 10 minutes

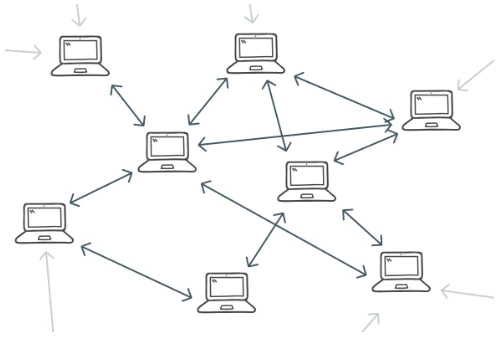
nVersion
hashPrevBlock
hashMerkleRoot
nTime
nBits
nNonce

- Every 2016 blocks, the target *T* is recalculated
- Let t_{sum} = Number of seconds taken to mine last 2016 blocks, then the target T_{new}

$$T_{\text{new}} = \frac{t_{\text{sum}}}{2016 \times 10 \times 60} \times T$$

- Recall that probability of success in single try is $\frac{T+1}{2^{256}}$
- If t_{sum} = 2016 × 8 × 60, then $T_{\text{new}} = \frac{4}{5} \times T$
- If t_{sum} = 2016 × 12 × 60, then $T_{\text{new}} = \frac{6}{5} \times T$

The Bitcoin P2P Network



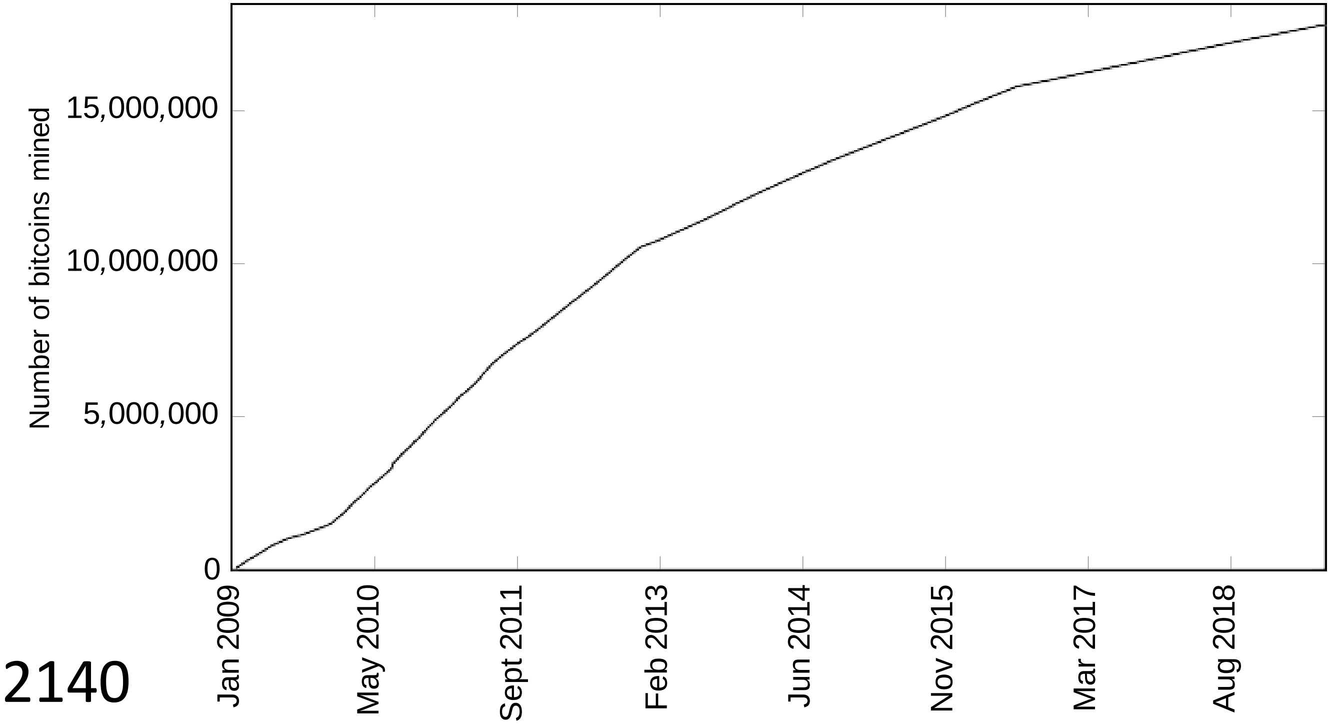
- Three types of nodes
  - Full Node
    - A full node stores a copy of blockchain on their local storage
  - Miner
    - A miner is a full node that takes part in adding blocks to the blockchain
  - Simple Payment Verification (SPV) Node
    - A SPV node only stores the block header
    - They contact full nodes when additional information about transactions is required

Bitcoin Blockchain Explorers

- Web interfaces to view current blockchain state
  - https://www.blockstream.info
  - https://www.blockchain.com/explorer
  - https://btc.com/
- Demo checklist
  - Address generation at https://www.bitaddress.org
  - Brainwallet generation at https://brainwalletx.github.io

Bitcoin Supply

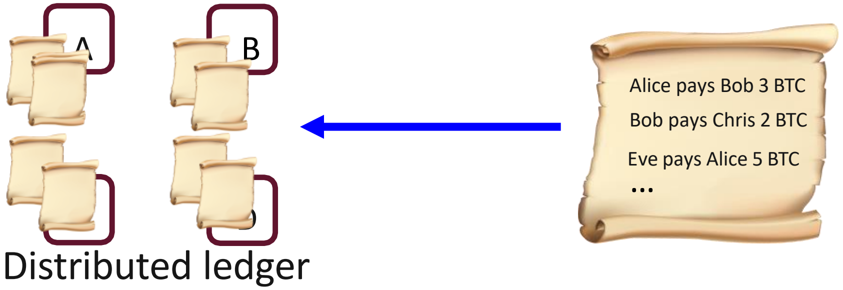
- The block subsidy was initially 50 BTC per block
- Halves every 210,000 blocks ≈ 4 years
  - 25 BTC in Nov 2012, 12.5 BTC in July 2016, and 6.25 BTC in May 2020
- Total Bitcoin supply is 21 million



- The last bitcoin will be mined in 2140

Blockchain – A High-level View

Applications



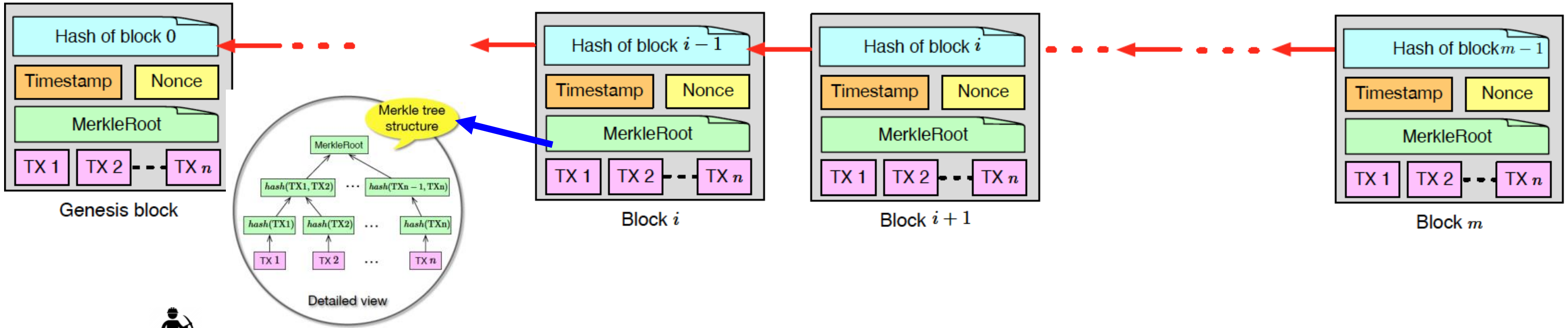
Alice pays Bob 3 BTC

Bob pays Chris 2 BTC

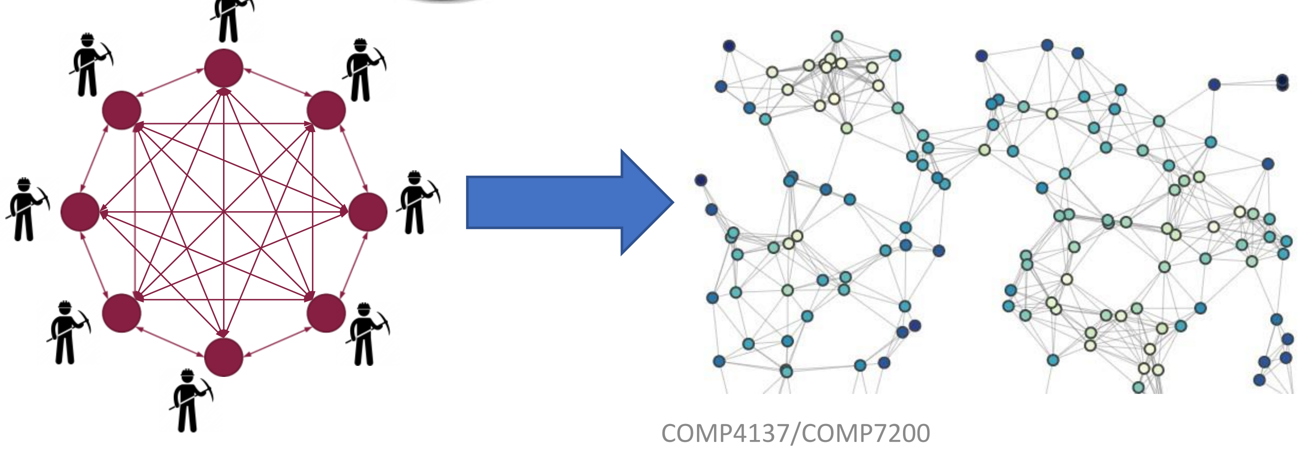
Eve pays Alice 5 BTC

…

Data structure



Consensus



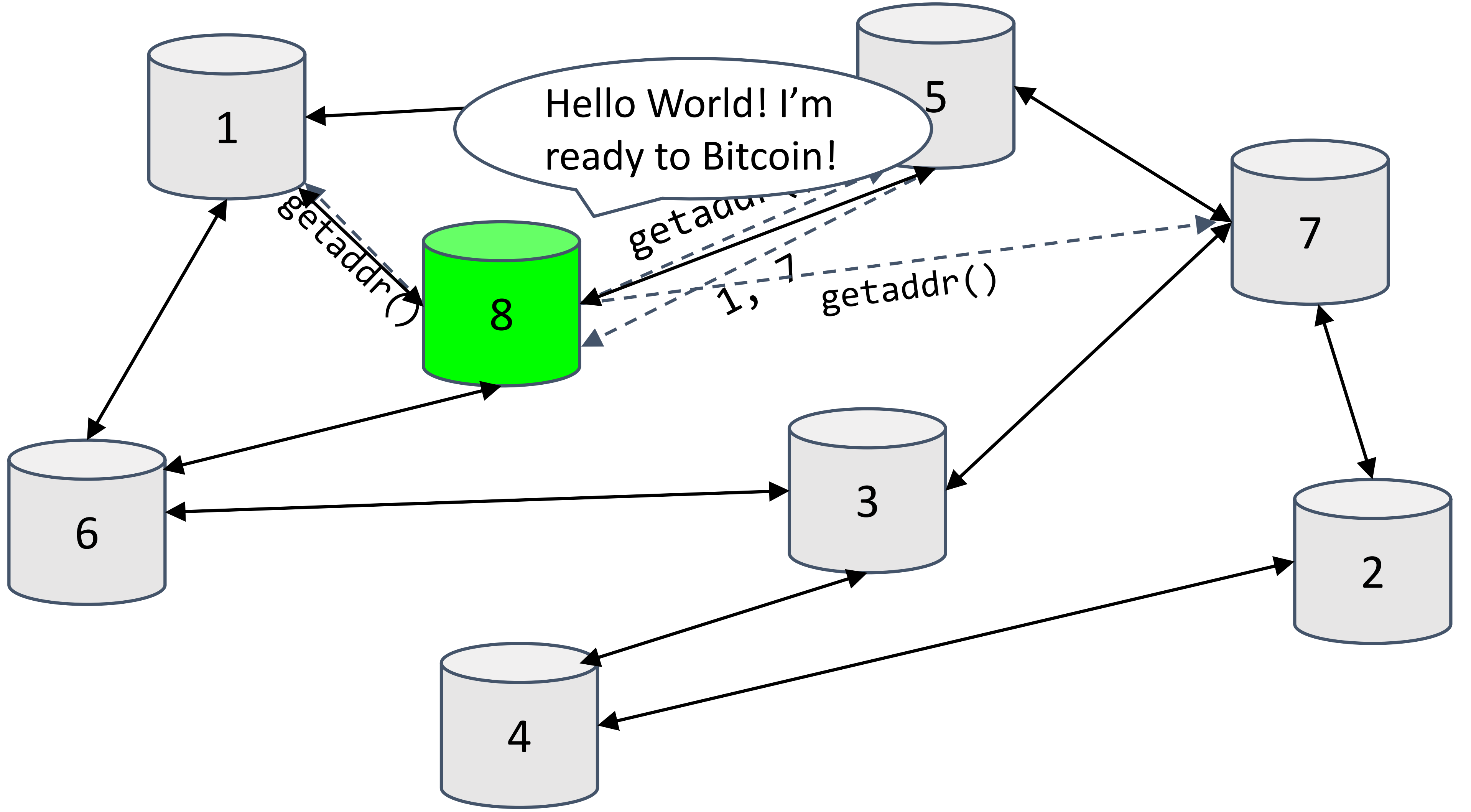
Bitcoin Testnet Transactions

- Each cryptocurrency has a mainnet and one or more testnets
- Bitcoin Testnet
  - https://live.blockcypher.com/btc-testnet/
- Testnet Address Generator
  - https://bitcoinpaperwallet.com/bitcoinpaperwallet/generate-wallet.html?design=alt-testnet
- Testnet faucet 1
  - https://coinfaucet.eu/en/btc-testnet/
- Testnet faucet 2
  - https://bitcoinfaucet.uo1.net
- Mycelium Testnet Wallet Mobile APP

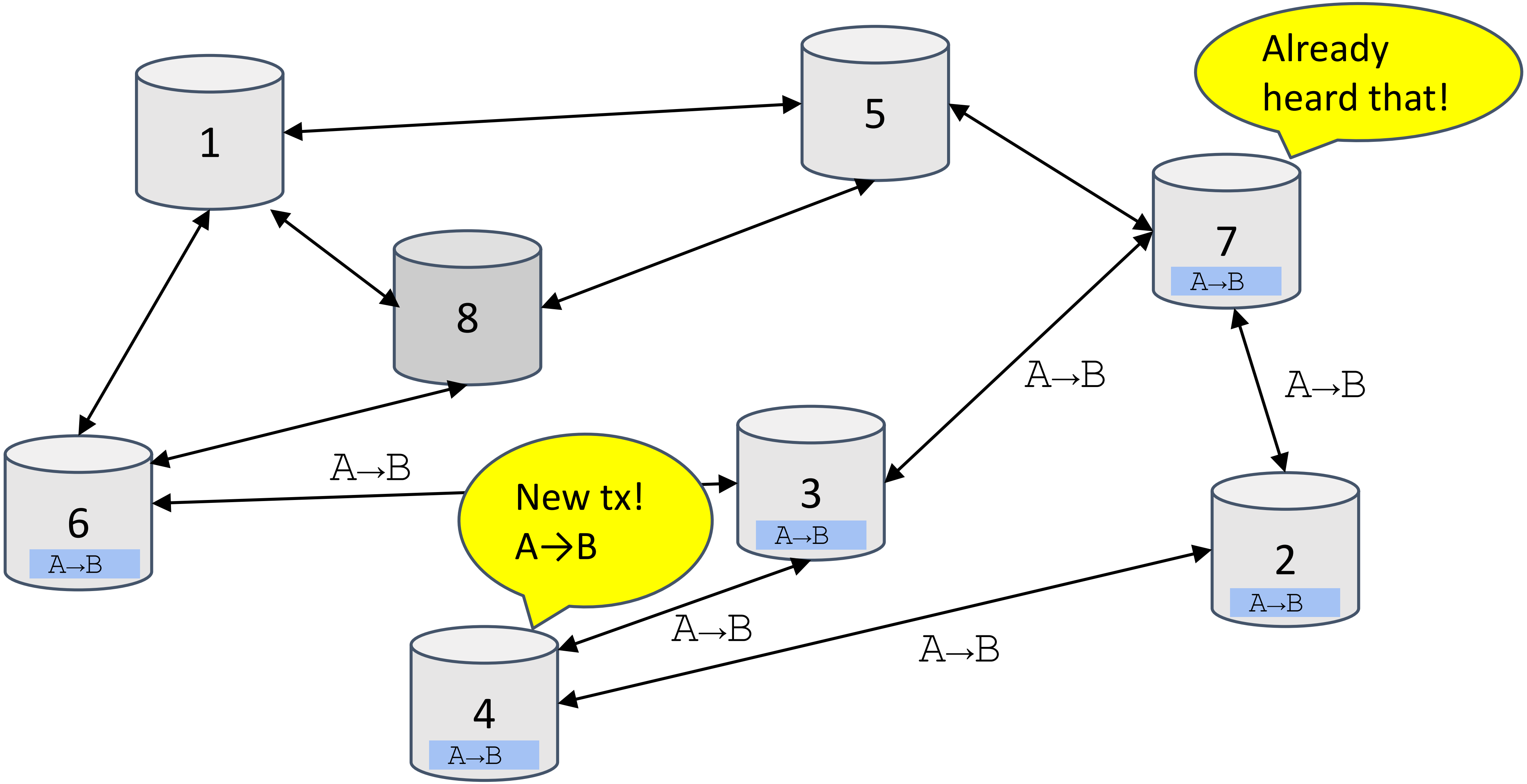
Bitcoin P2P network

- Ad-hoc protocol (runs on TCP port 8333)
- Ad-hoc network with random topology
- All nodes are equal
- New nodes can join at any time
- Forget non-responding nodes after 3 hr

Joining the Bitcoin P2P network



Transaction propagation (flooding)

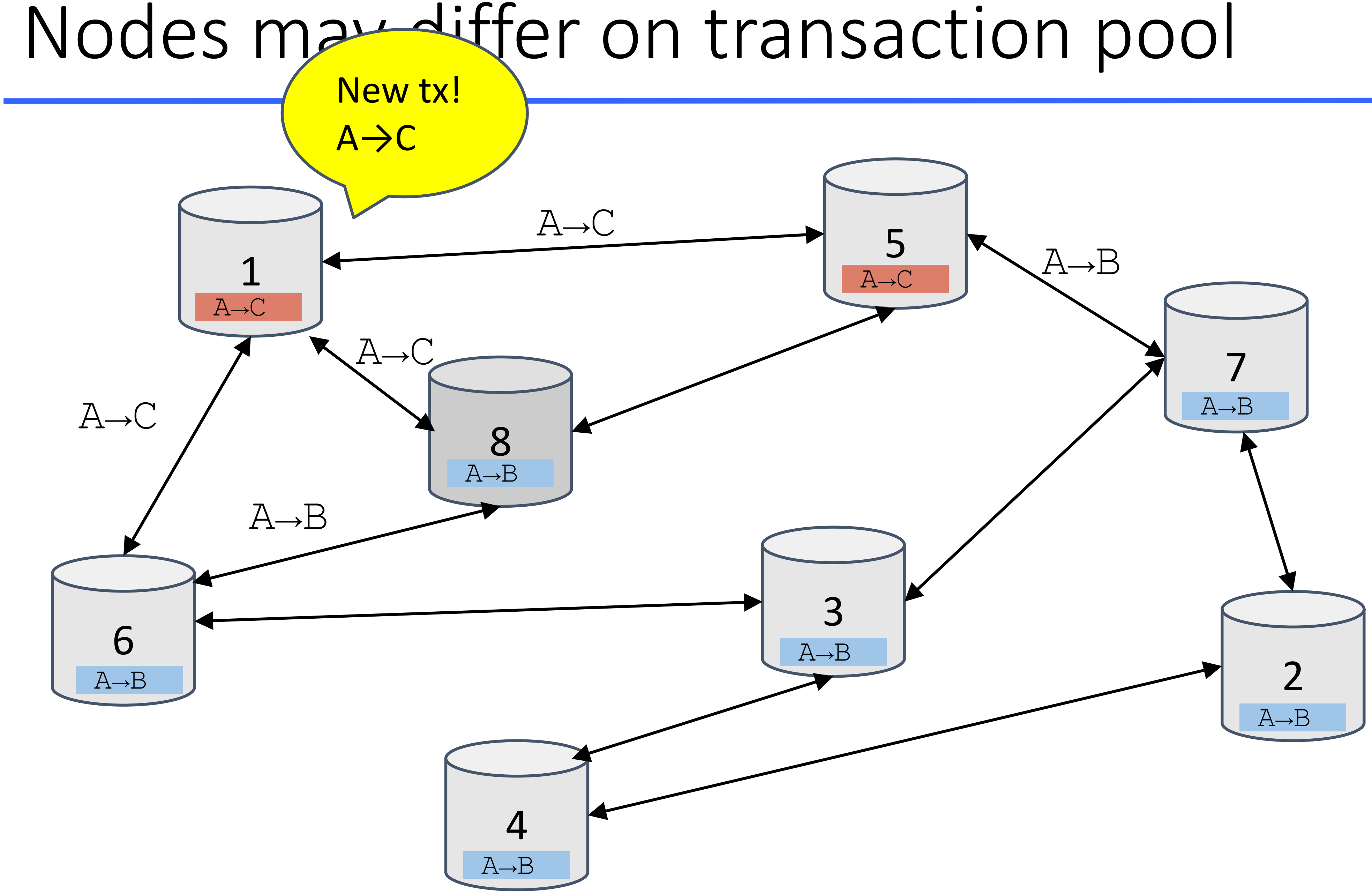


Should I relay a proposed transaction?

- Transaction valid with current block chain (default)
  - Run script for each previous output being redeemed and ensure that script returns true!
- Script matches a whitelist
  - Avoid unusual scripts
- Haven’t seen before
  - Avoid infinite loops
- Doesn’t conflict with others I’ve relayed
  - Avoid double-spends

Sanity checks only...
Well-behaving nodes implement them!
Some nodes may ignore them!

Nodes may differ on transaction pool

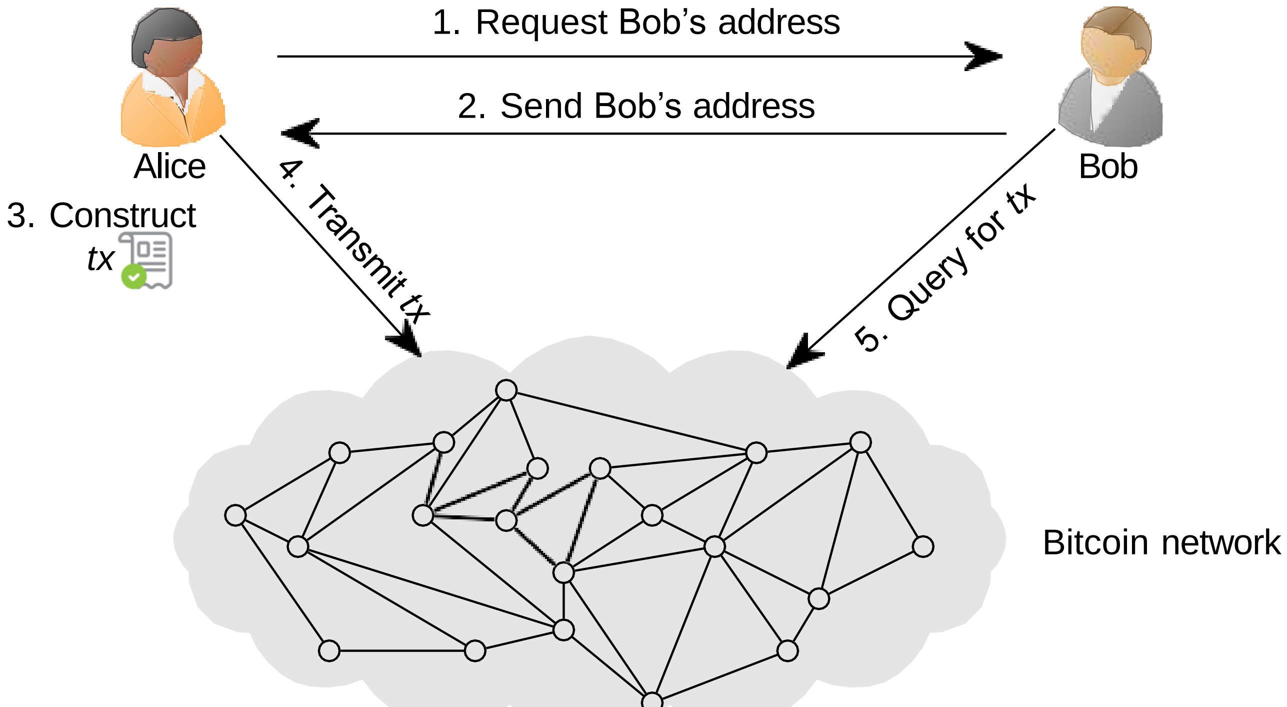


Outline

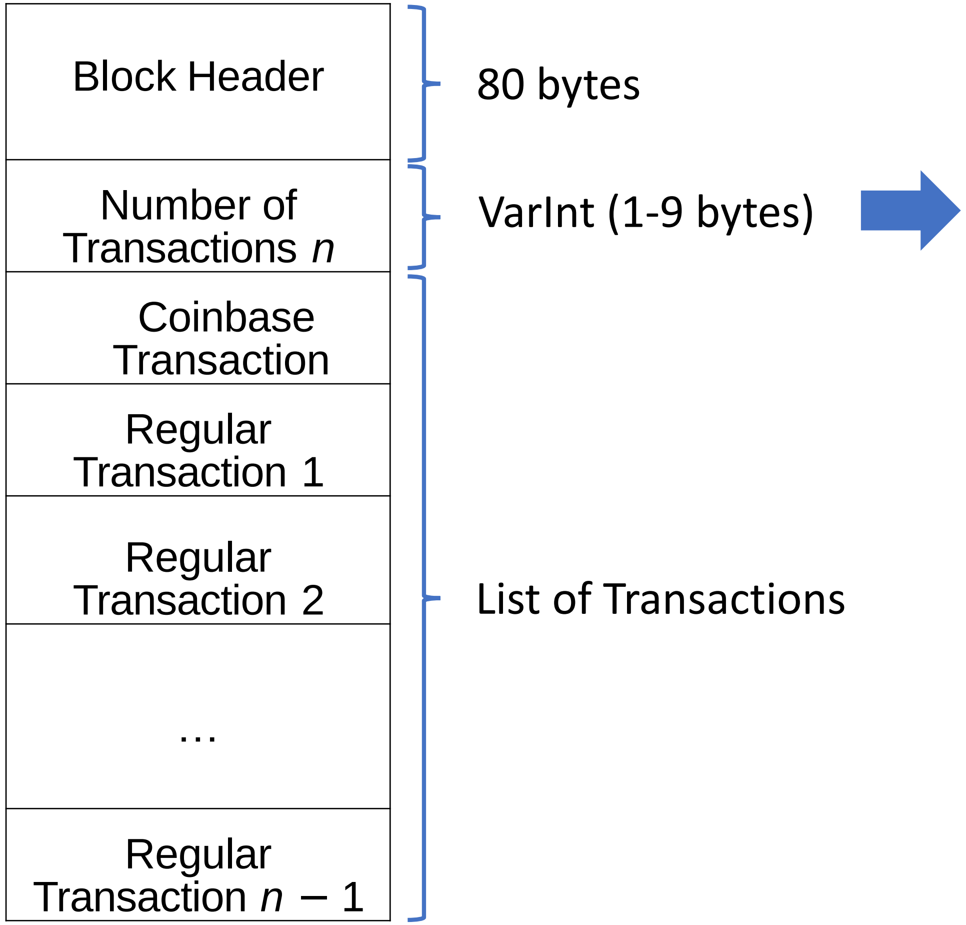
- Bitcoin Block Format
  - Header
  - Body
- Bitcoin Consensus
  - Mining
  - Target Threshold
- Bitcoin Transactions
  - Transaction Format
  - Script

Bitcoin Payment Workflow

- Merchant Bob shares address out of band (not using Bitcoin P2P)
- Customer Alice broadcasts transaction *tx*, which pays the address
- Miners collect broadcasted transactions into a candidate block
- One of the candidate blocks containing *tx* is mined
- Bob waits for confirmations on *t* before providing goods



Block Format



Block Header

Number of Transactions n

Coinbase Transaction

Regular Transaction 1

Regular Transaction 2

…

Regular Transaction $n-1$

80 bytes

VarInt (1-9 bytes)

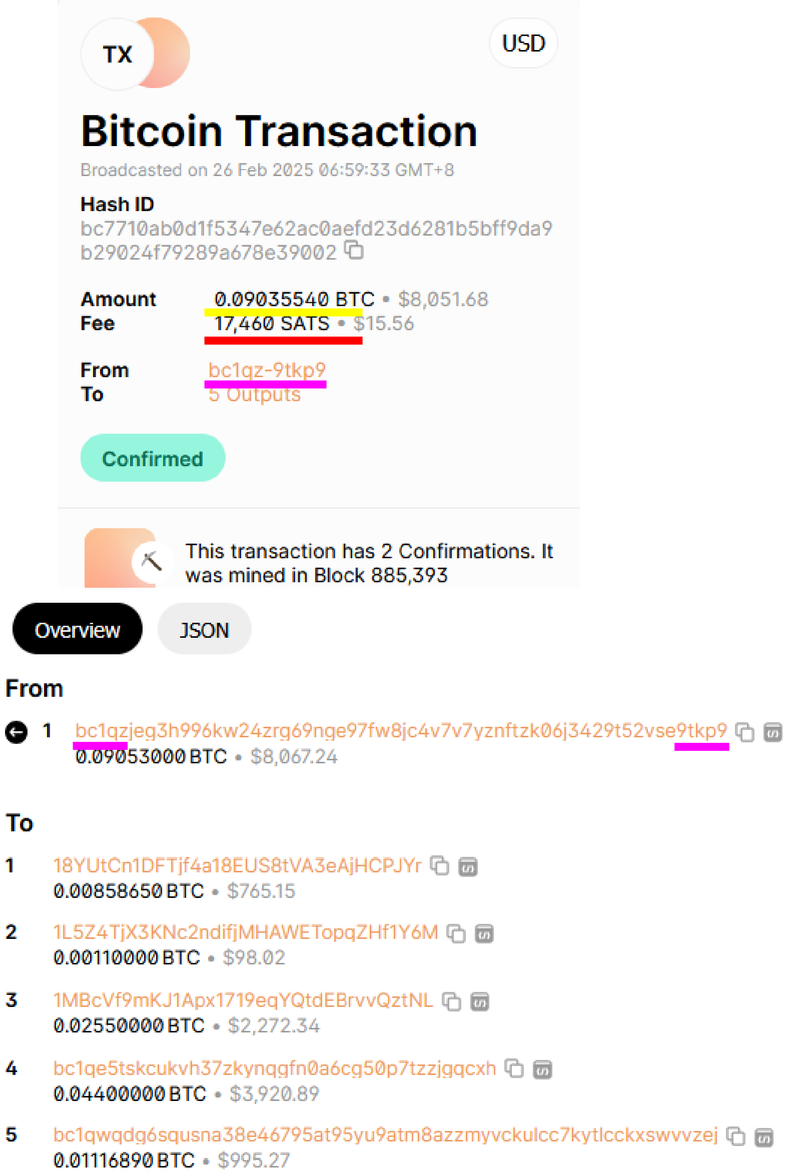
List of Transactions

Value of n	Size of VarInt (byte)	Encoding
0 - 252	1	n
253 – 2^{16} - 1	3	253||n
2^{16} – 2^{32} - 1	5	254||n
2^{32} – 2^{64} - 1	9	255||n

Bitcoin Transactions

- A Bitcoin transaction (Tx) encodes a transfer of bitcoins between entities.
- A destination of the transfer is called an **output**
  - A single Tx can have several outputs
  - Each output can serve as a source of bitcoins in a later Tx
- When previous Tx outputs are specified as the source of bitcoins in a transaction, they are called **inputs**
- A ***coinbase transaction*** has no input and at least one output.
  - There is no input because the source of bitcoins is not from a previous transaction, rather, it is from the block reward.

Examples



TX

USD

Bitcoin Transaction

Broadcasted on 26 Feb 2025 06:59:33 GMT+8

Hash ID
bc7710ab0d1f5347e62ac0aefd23d6281b5bff9da9b29024f79289a678e39002

Amount	0.09035540 BTC • $8,051.68
Fee	17,460 SATS • $15.56
From	bc1qz-9tkp9
To	5 Outputs

Confirmed

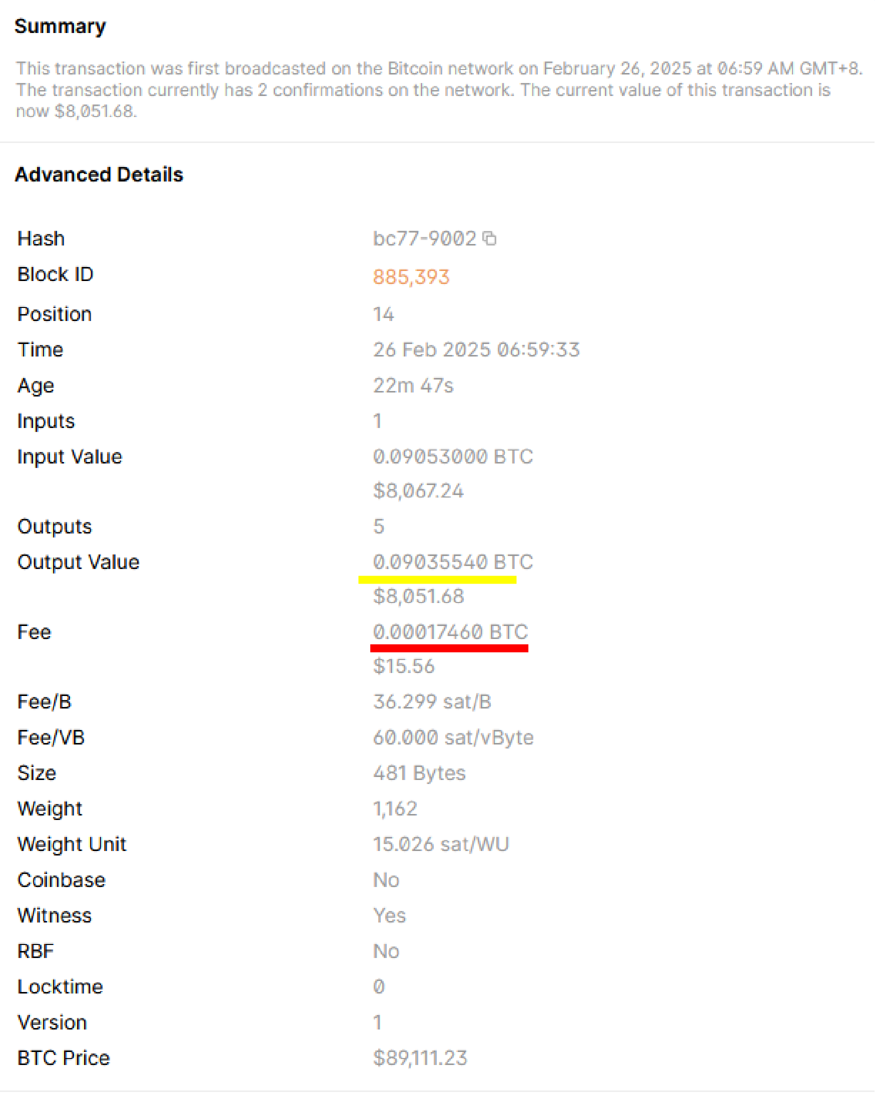
This transaction has 2 Confirmations. It was mined in Block 885,393

From

1. bc1qzjeg3h996kw24zrg69nge97fw8jc4v7v7yznftzk06j3429t52vse9tkp9
   0.09053000 BTC • $8,067.24

To

1. 18YUtCn1DFTjf4a18EUS8tVA3eAjHCPJYr
   0.00858650 BTC • $765.15
2. 1L5Z4TjX3KNc2ndifjMHAWETopqZHf1Y6M
   0.00110000 BTC • $98.02
3. 1MBcVf9mKJ1Apx1719eqYQtdEBrvvQztNL
   0.02550000 BTC • $2,272.34
4. bc1qe5tskcukvh37zkynqgfn0a6cg50p7tzzjgqcxh
   0.04400000 BTC • $3,920.89
5. bc1qwqdg6squsna38e46795at95yu9atm8azzmyvckulcc7kytlcckxswvvzej
   0.01116890 BTC • $995.27



Summary

This transaction was first broadcasted on the Bitcoin network on February 26, 2025 at 06:59 AM GMT+8. The transaction currently has 2 confirmations on the network. The current value of this transaction is now $8,051.68.

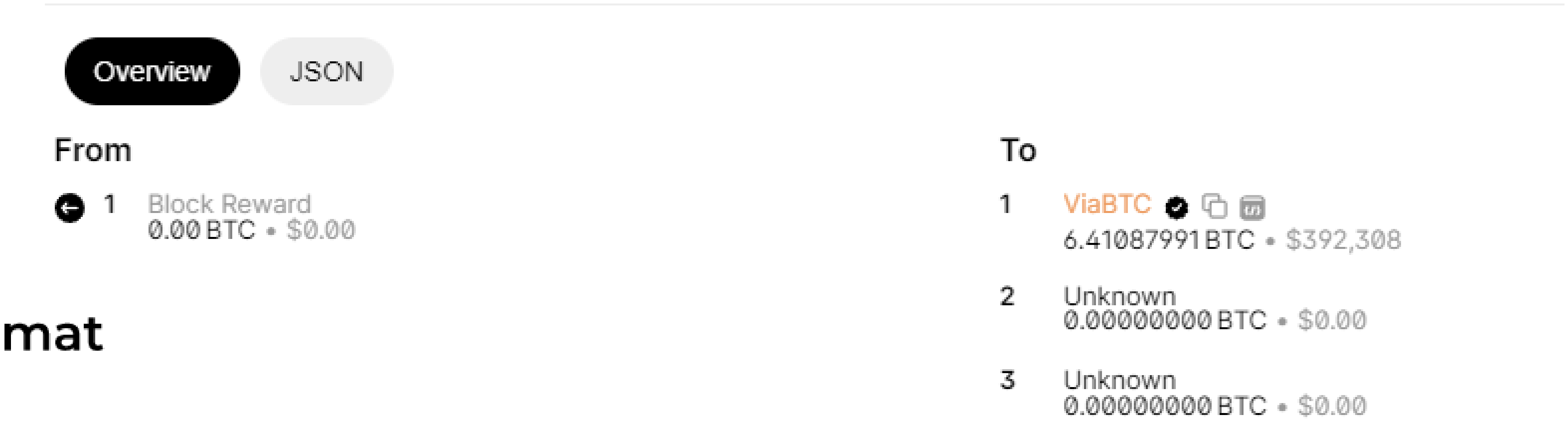
Advanced Details

Hash	bc77-9002
Block ID	885,393
Position	14
Time	26 Feb 2025 06:59:33
Age	22m 47s
Inputs	1
Input Value	0.09053000 BTC $8,067.24
Outputs	5
Output Value	0.09035540 BTC $8,051.68
Fee	0.00017460 BTC $15.56
Fee/B	36.299 sat/B
Fee/VB	60.000 sat/vByte
Size	481 Bytes
Weight	1,162
Weight Unit	15.026 sat/WU
Coinbase	No
Witness	Yes
RBF	No
Locktime	0
Version	1
BTC Price	$89,111.23

https://www.blockchain.com/explorer/ (and many other sites)

Examples (coinbase)

- https://www.blockchain.com/explorer/transactions/btc/04f535a736834ce1b711fd1fb94ca418e470d65b98bd8b3dfd163b8fc8bac026



Overview JSON

From

1 Block Reward
0.00 BTC • $0.00

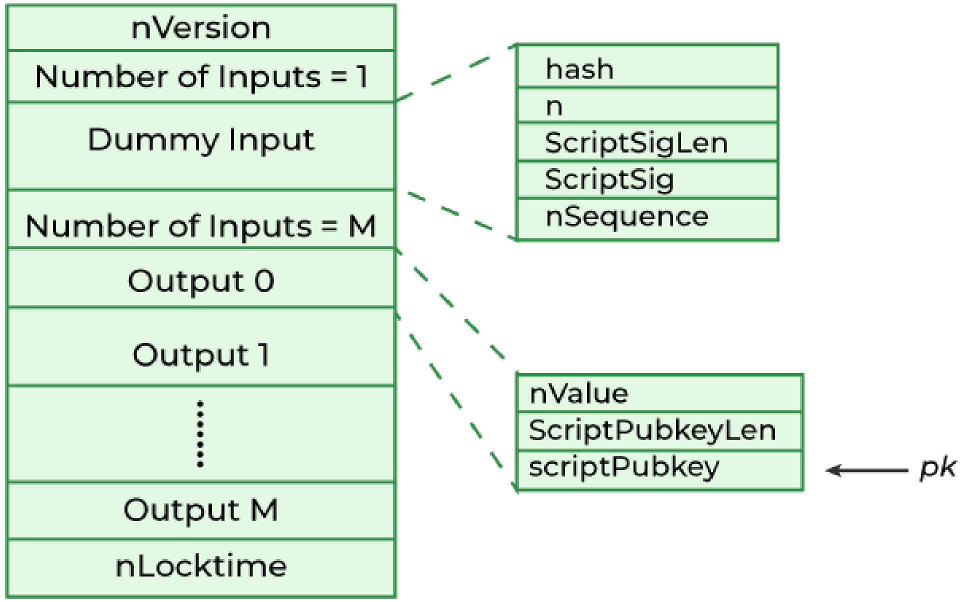
To

1 ViaBTC
6.41087991 BTC • $392,308

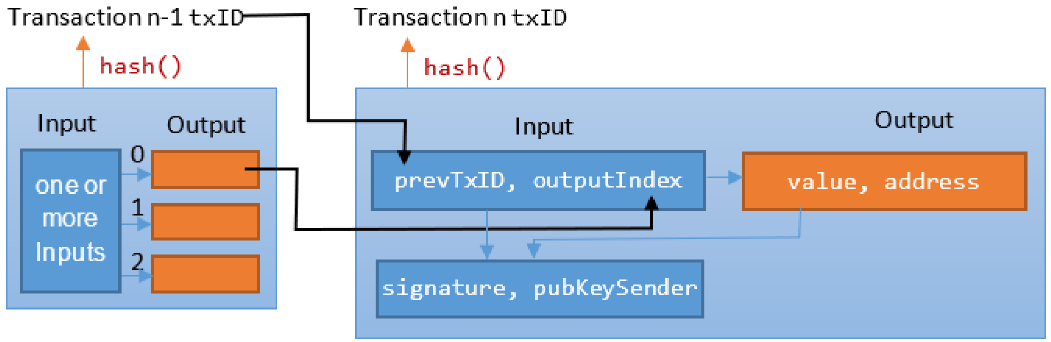
2 Unknown
0.00000000 BTC • $0.00

3 Unknown
0.00000000 BTC • $0.00

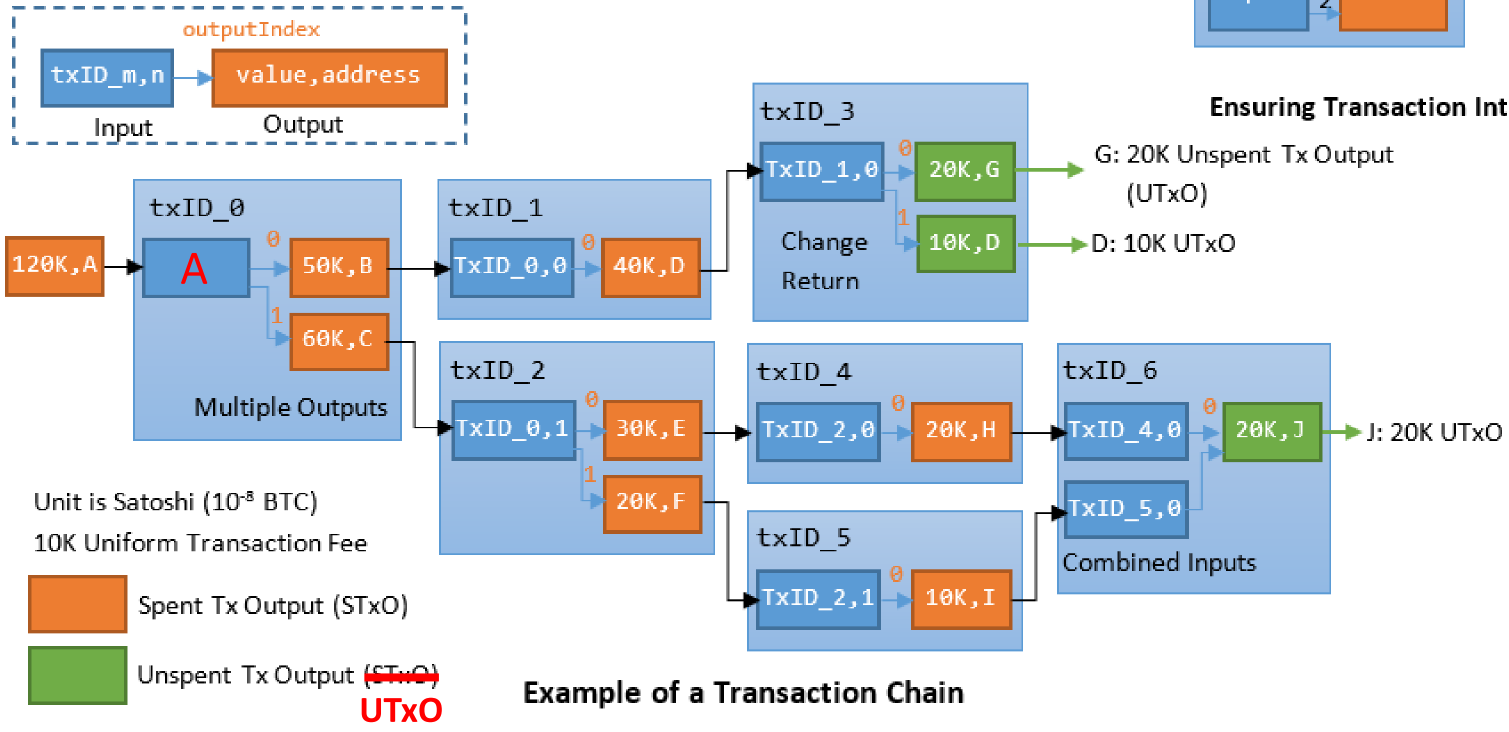
Coinbase Transaction Format



Examples



Ensuring Transaction Integrity (Blockchain Immutability) via Chaining of Hashes

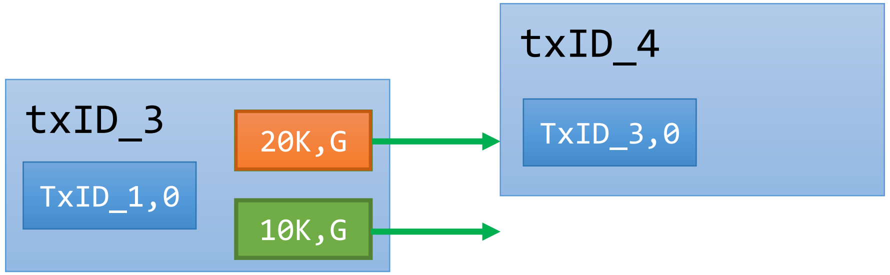


Example of a Transaction Chain

- An output contains the value to be transferred and the recipient's address (or public key)
  - Multiple outputs are allowed in a transaction
- An input refers to a previous *unspent transaction output* (UTxO)
- Multiple inputs are allowed in a transaction

Bitcoin Ownership (1)

- When an output of a previous transaction is "unlocked" by the input of a later transaction, all the bitcoins in this output need to be spent
  - A transaction output can be in only one of the two states, namely, spent or unspent
- **Unspent transaction outputs (UTXOs)**
  - Refers to outputs in Tx which have not been referred by the inputs of later transactions

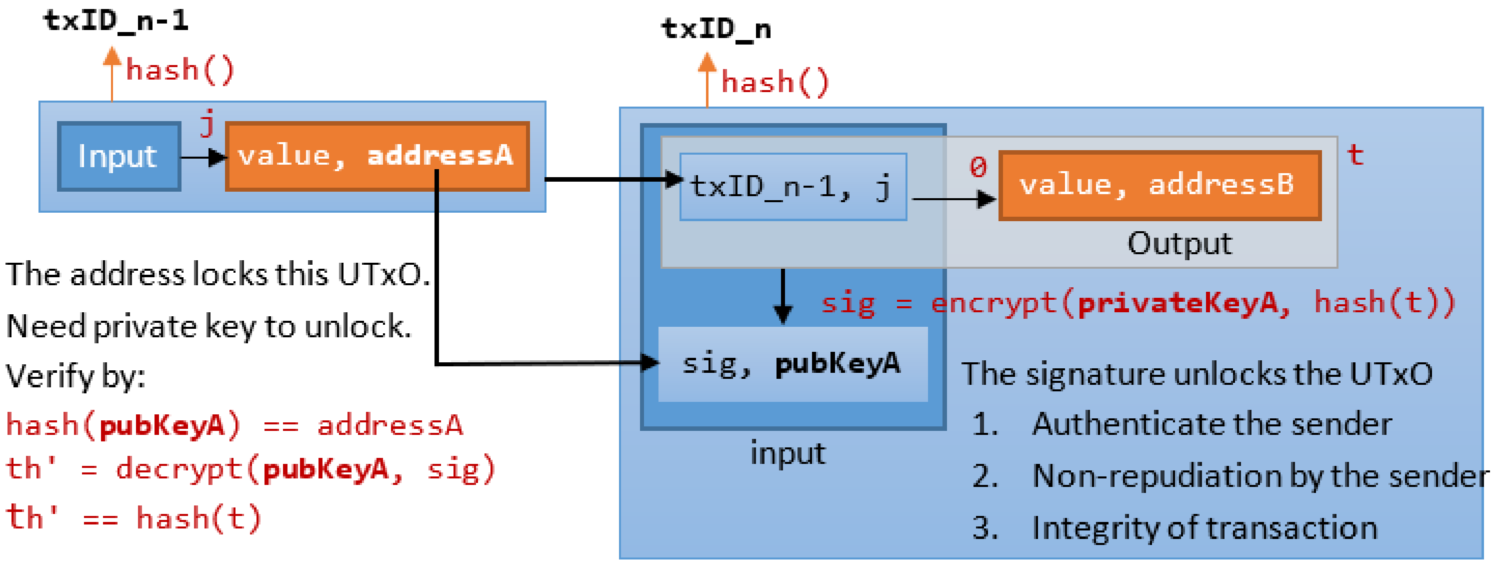


Bitcoin Ownership (2)

- When a new block is added, the output of the coinbase transaction is a UTXO
- Every regular transaction in the new block unlocks UTXOs from the previous blocks and creates new UTXOs
- The unlocked outputs in the previous Tx are not UTXOs
    - The set of UTXOs changes with every new block
- UTXO model is different from the traditional account model in the bank
    - Provide anonymity

Bitcoin Ownership (3)

- The set of all UTXOs that an entity can unlock can be thought of as bitcoins owned by that entity
- During a fork, different nodes may consider different branches and thus the UTXO set will differ across nodes with different local copies
- The UTXO set will be the same when the local copies become the same (after the fork is resolved)



The address locks this UTxO.
Need private key to unlock.
Verify by:

```
hash(pubKeyA) == addressA
th' = decrypt(pubKeyA, sig)
th' == hash(t)
```

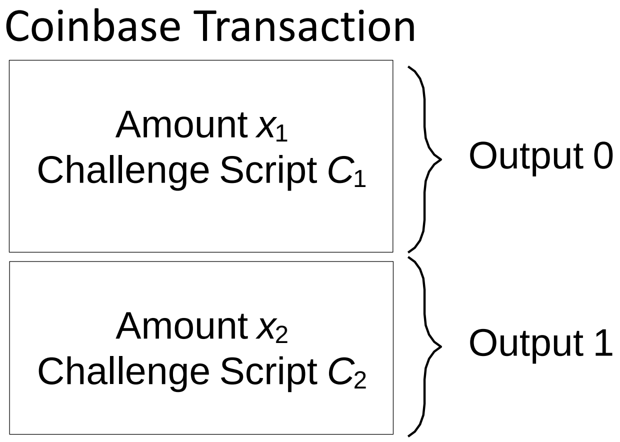
The signature unlocks the UTxO

1. Authenticate the sender
2. Non-repudiation by the sender
3. Integrity of transaction

Unlock the UTxO by Signing to Certify its Consumption

Coinbase Transaction

- Each output in the ***coinbase transaction*** contains two items:
  - Amount of bitcoins
  - A script which specifies the conditions under which the bitcoins associated with this output can be spent
- The script in an output can be viewed as a challenge.
  - An entity which provides a satisfactory response can transfer the bitcoins associated with the output



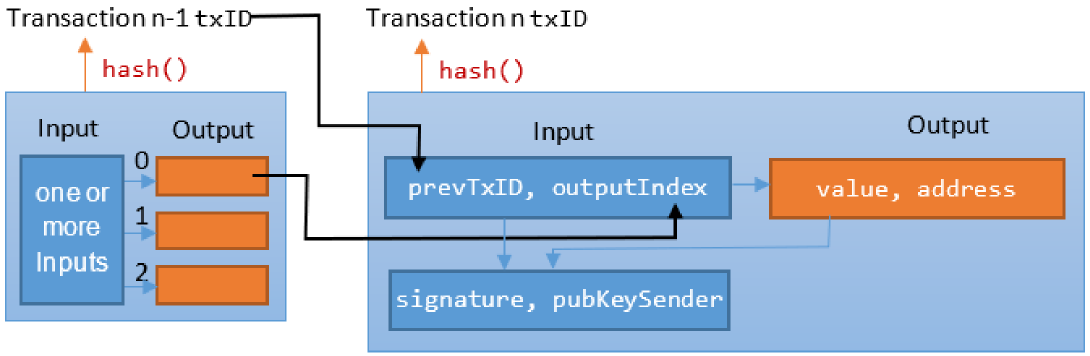
Coinbase Transaction

- The sum of the amounts in all outputs of the coinbase transaction must be less than or equal to the block reward
    - If less, some of the bitcoin is not spendable
- So, usually the sum of amounts of all outputs of a coinbase transaction is equal to the block reward

- Coinbase Transaction demo
    - https://andersbrownworth.com/blockchain/coinbase

Regular Transaction

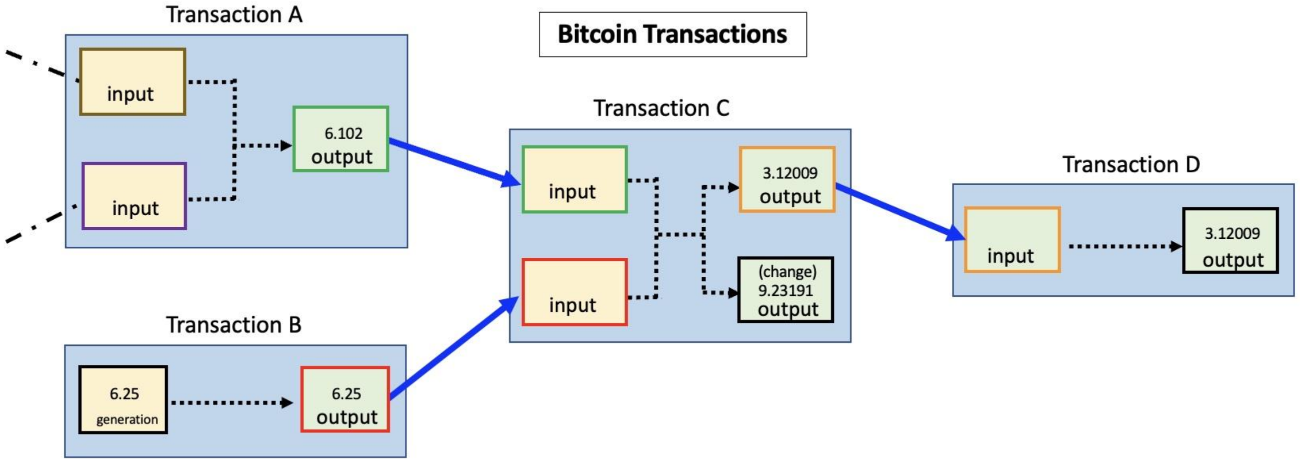
- A **regular transaction** spends the bitcoins earned in a coinbase transaction or received from a regular transaction.
- Each regular transaction must have at least one input and one output.
- The **outputs** in a regular transaction have the same format as the outputs in a coinbase transaction



Ensuring Transaction Integrity (Blockchain Immutability) via Chaining of Hashes

Regular Transaction Input

- One **input** includes following information:
  - ***Transaction Identifier (TxID) of a previous transaction*** on the blockchain. TxID is the double SHA-256 hash of the transaction
  - ***The index of an output in the previous transaction***, starting from 0.
  - ***A response script*** which satisfies the condition required to spend the bitcoins in the output
- The inputs don't specify the amount of Bitcoins to be spent.
- If an input refers to an output of a previous Tx, all BTCs associated with that output must be spent in the Tx.



Regular Transaction Fee

- Suppose a regular transaction has *N* inputs and *M* outputs
- Let x_1, x_2, …, x_N be the bitcoins associated with the *N* inputs (i.e., *N* outputs of previous transactions)
- Let y_1, y_2, …, y_M be the bitcoin associated with the *M* outputs
- Then, the transaction fee denoted by *R* is defined as

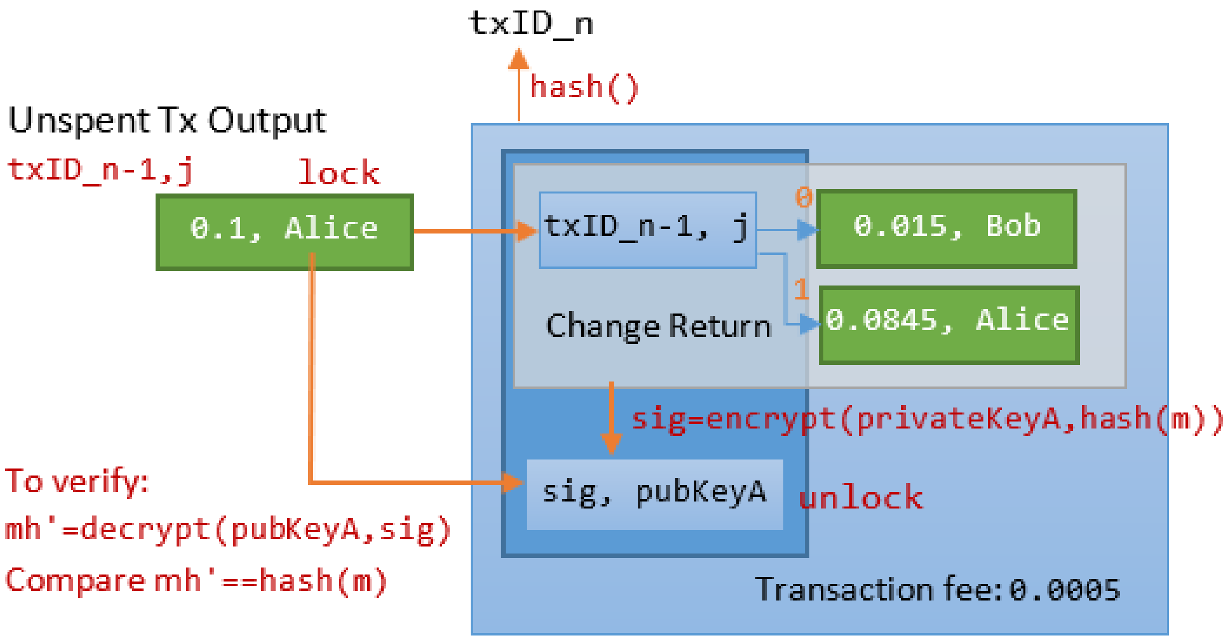
$$R = \sum_{i=1}^{N} x_i - \sum_{j=1}^{M} y_j$$

where $\sum_{i=1}^{N} x_i \geq \sum_{j=1}^{M} y_j$.

Regular Transaction Fee Rate

- Miners aim to maximize their block reward
- Block subsidy is fixed
- Transaction fee depends on the transaction miner chooses to include in the block
    - High transaction fee
    - Small transaction size
    - Transaction fee per byte (or fee rate) is the factor for them to be considered

Example of writing a new transaction

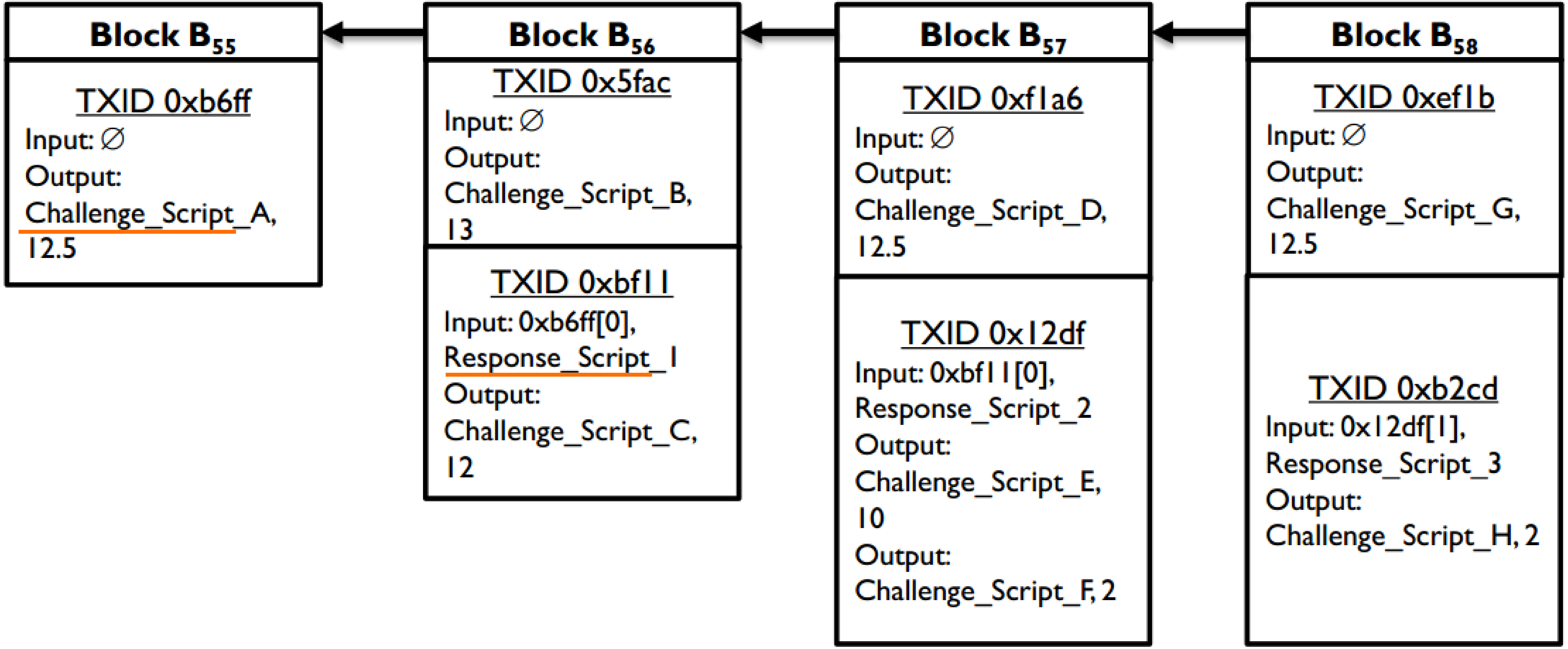


An Example of Writing a New Transaction

Input: 0.1
Output: 0.015+0.0845=0.0995

Fee: 0.0005 = Input - Output

Bitcoin Script



Block B_{55}

TXID 0xb6ff
Input: ∅
Output:
Challenge_Script_A, 12.5

Block B_{56}

TXID 0x5fac
Input: ∅
Output:
Challenge_Script_B, 13

TXID 0xbf11
Input: 0xb6ff[0], Response_Script_1
Output:
Challenge_Script_C, 12

Block B_{57}

TXID 0xf1a6
Input: ∅
Output:
Challenge_Script_D, 12.5

TXID 0x12df
Input: 0xbf11[0], Response_Script_2
Output:
Challenge_Script_E, 10
Output:
Challenge_Script_F, 2

Block B_{58}

TXID 0xef1b
Input: ∅
Output:
Challenge_Script_G, 12.5

TXID 0xb2cd
Input: 0x12df[1], Response_Script_3
Output:
Challenge_Script_H, 2

Bitcoin Script

- Response and Challenge scripts are encoded using a special scripting language developed by Bitcoin
  - Stored in `scriptSig` and `scriptPubkey` fields of a transaction

- This language is simply called Script
  - It is a stack-based language.
  - It is not a general-purpose language and its goal is to support bitcoin transactions.

Standard Transactions

- To prevent a Denial-of-Service (DoS) attack, nodes in the Bitcoin network will only relay transactions containing ***challenge scripts of some pre-defined forms***.

- They are called standard transactions.

Standard Transactions

- Pay to Public Key (P2PK)
  - Public key as payment destination
- Pay to Public Key Hash (**P2PKH**)
  - Hash of public key as payment destination
- M-of-N multi-signature
  - Response script provides signatures created using any m out of the n private keys
- Pay to Script Hash (P2SH)
  - Hash of a script as payment destination
- Null Data
  - Mainly used to timestamp data

Summary

- Bitcoin’s blockchain prevents double spending and tampering
- Secure only if nobody controls 50% or more of network hashrate
- Mining difficulty adjusted to regulate coin supply
- Miners incentivized by block reward
- Block subsidy halves every four years to limit total coin supply
- Bitcoin addresses are shared over the Internet
- Transactions paying these addresses are broadcasted on the Bitcoin network

References

- Saravanan Vijayakumaran, “An Introduction to Bitcoin”
- Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, “Bitcoin and Cryptocurrency Technologies”
- Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”
- Bitcoin Charts
  - https://www.blockchain.com/charts
- Bitmain Mining Rigs
  - https://shop.bitmain.com