Blockchain Boot Camp

Prof. Kirk W. Cameron
ADH for Research and Engagement
Virginia Tech
Computer Science

What this Boot Camp is not…

- A comprehensive course on Blockchain, crypto, finance, algorithms,…
- A guide to “how to make money from cryptocurrency”
- An endorsement of any particular cryptocurrency, company, technology related to blockchain
- An endorsement of any particular politics, government, lifestyle associated with cryptocurrency, blockchain, macro or micro economics, etc.
- A place for you to raise money for your blockchain startup

What this Boot Camp is...

- (Mostly) Technical discussion of the breadth and potential for blockchain tech
- An intro to Blockchains
- Coverage of important systems/C++ constructs
- Exposure to local industry experts and emergent technologies
- A place to link up with others interested in blockchain
- An encouragement for you to learn more on your own, with friends, at Virginia Tech, and beyond
- And some exciting news...

You should be able to…

- Understand the importance of Blockchain technology
- Understand how transactions are processed and validated
- Comprehend basic cryptographic concepts relevant to blockchain technologies
- Discuss the breadth of use-cases of the blockchain
- Learn where and how to participate in blockchain development activities
- Understand how to use EOSIO to create your own blockchain applications

What we assume about your background…

- Many students with background/coursework/experience in computing (mostly CS/ECE)
  - back-end development (Linux, Python, C++)
  - front-end development (JavaScript, NodeJS)
- BUT, some folks from many other disciplines
  - Industrial, BIT, Finance, Aerospace, Biomedical
  - Math, MIT, ME, Physics, Stats
- And some professionals from Region 2 (New River Valley)
  - Welcome, part of GO Virginia grant to Virginia Tech!
- Some time set aside for mingling
  - Great ideas come from everywhere!
  - Chat during breakfast, coffee, lunch
  - Chance to recruit folks to work with you

Boot Camp Agenda (Jan 18, 2020)

- 8:30 a.m. – 9:00 a.m. Breakfast
- 9:00 a.m. – 9:30 a.m Welcome from Prof. Cameron (and Dan Larimer)
- 9:30 a.m. – 10:00 a.m. Intro to Blockchain
- 10:00 a.m. – 10:30 a.m. COFFEE BREAK
- 10:30 a.m. – 12:00 a.m. C++ and WASM Intro (Prof. Godmar Back)
- 12:00 a.m. – 12:30 p.m. LUNCH
- 12:30 p.m. – 2:00 p.m. EOSIO Intro (Dr. ”Bucky” Kittenger)
- 2:00 p.m. – 2:15 p.m. BREAK
- 2:15 p.m. – 3:15 p.m. Blockchain DAPP / Web App Development (Jeff)
- 3:15 p.m. – 3:30 p.m. Closing and Networking

A special announcement

Dan Larimer, CTO Block.One

Notes on Challenge

Intro to Blockchain

Skillset needed for Blockchain Development

1. High performance computing
2. Computer performance engineering
3. Systems programming
4. Networking P2P, asynchronous
5. Cryptography
6. Databases
7. Game theory
8. Economics
9. Governance
10. Political Science
11. Law

Skillset needed for Blockchain Development

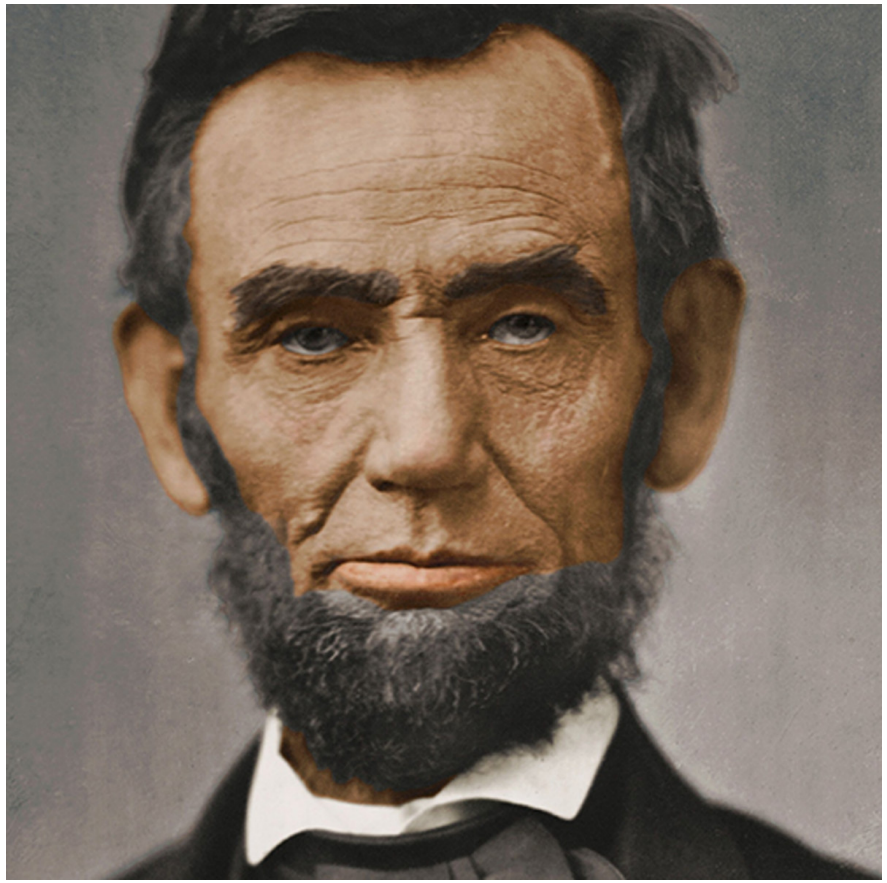
1. High performance computing
2. Computer performance engineering
3. Systems programming
4. Networking P2P, asynchronous
5. Cryptography
6. Databases
7. Game theory
8. Economics
9. Governance
10. Political Science
11. Law

Collaboration is in your future: 1/3 + 1/3 + 1/3

Some caveats before we go further…

- This is an intro from a distributed systems person.
  - I have my expertise. I have my biases. Deeper details to come.
  - I have a room full of experts across the other areas.
- Target audience skill set balkanized:
  - Blockchain, crypto, systems, none of these, etc.
- There are (or will be) semester-long courses on these topics.
- Goal here is coverage of blockchain basics.

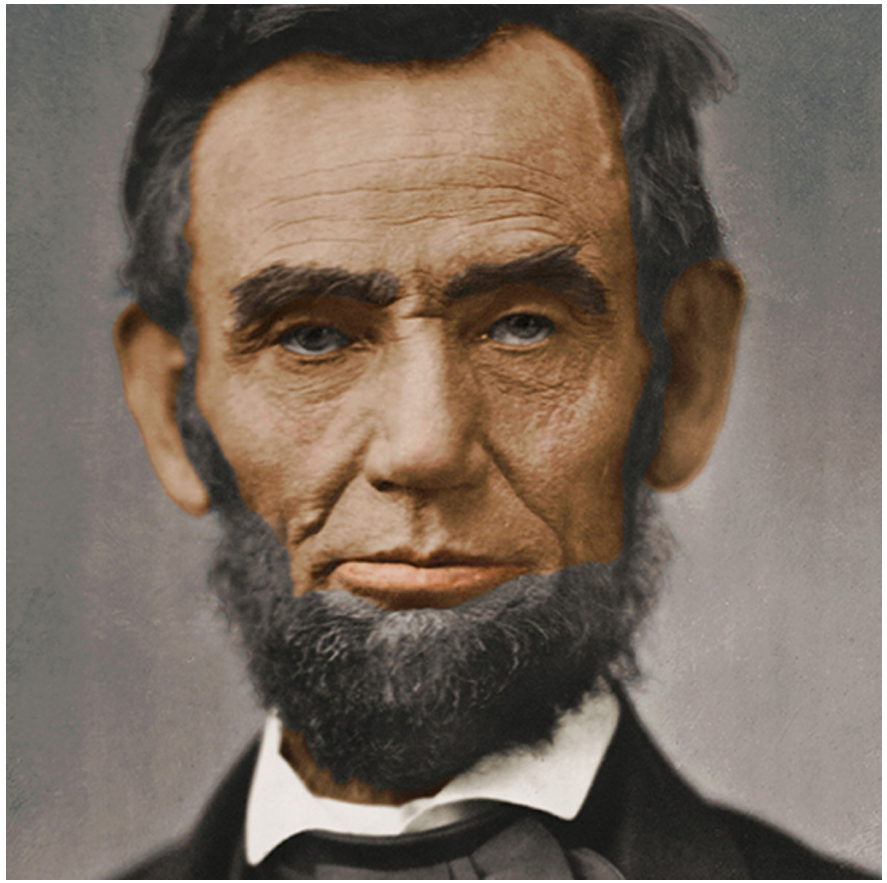
Why blockchain?



"Everything you read on the Internet is TRUE!"

-Abraham Lincoln
April 12, 1864

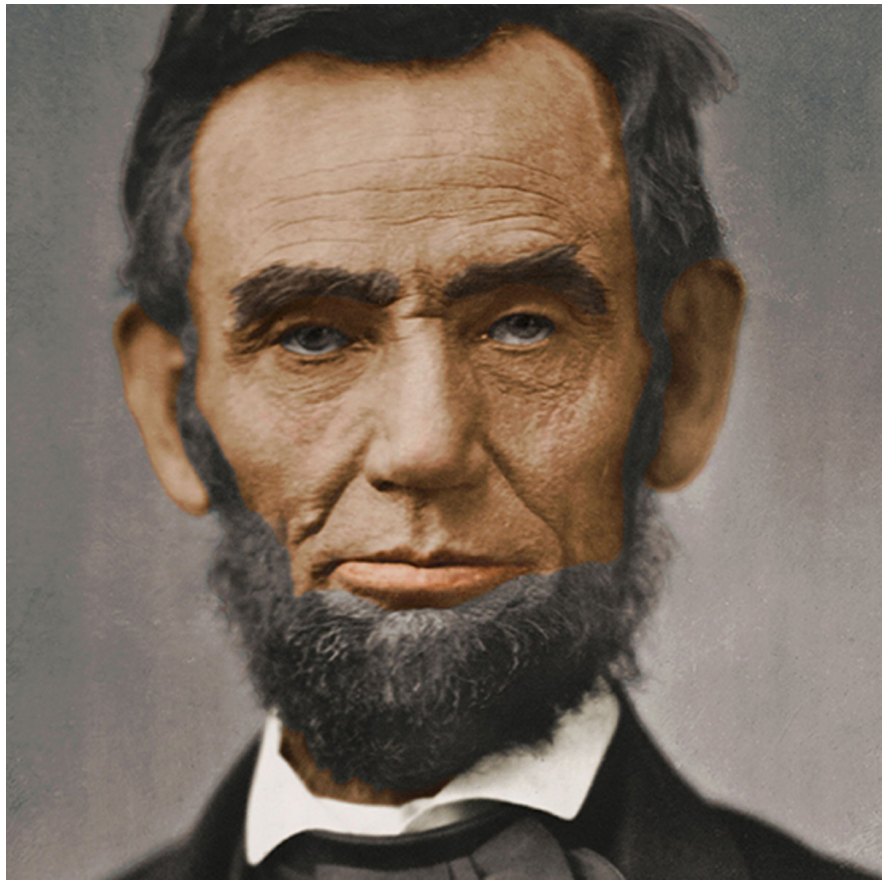
Why blockchain?



"Everything you read on the Internet is FALSE!"

-Abraham Lincoln
April 12, 1864

Why blockchain?

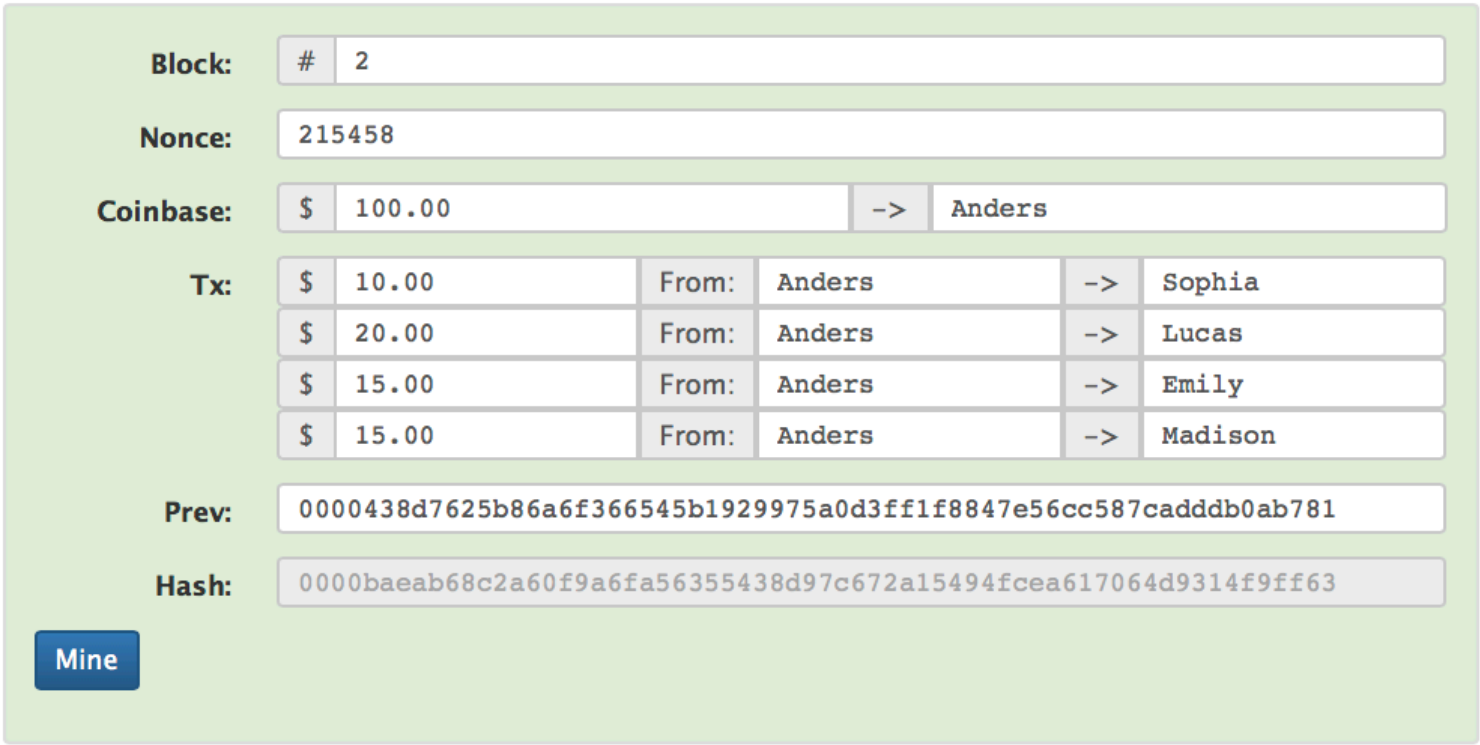


"Everything you read on the Internet is FALSE!"

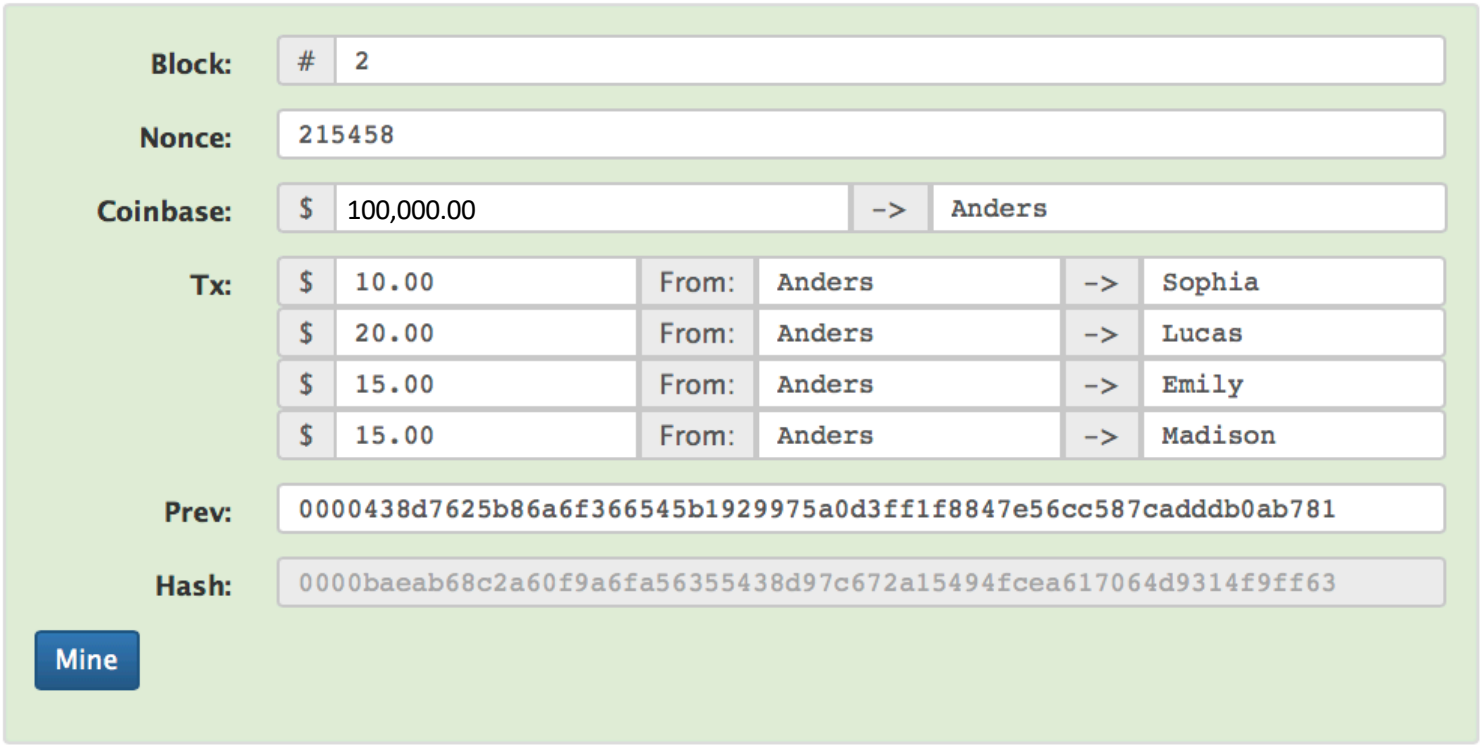
-Abraham Lincoln
April 17, 1864

In a world of digital (or virtual) "documents", how do we detect forgeries?

Who cares?


Block:
#
2
Nonce:
215458
Coinbase:
$
100.00
->
Anders
Tx:
$
10.00
From:
Anders
->
Sophia
$
20.00
From:
Anders
->
Lucas
$
15.00
From:
Anders
->
Emily
$
15.00
From:
Anders
->
Madison
Prev:
0000438d7625b86a6f366545b1929975a0d3ff1f8847e56cc587cadddb0ab781
Hash:
0000baeab68c2a60f9a6fa56355438d97c672a15494fcea617064d9314f9ff63
Mine

Who cares?


Block:
#
2
Nonce:
215458
Coinbase:
$
100,000.00
->
Anders
Tx:
$
10.00
From:
Anders
->
Sophia
$
20.00
From:
Anders
->
Lucas
$
15.00
From:
Anders
->
Emily
$
15.00
From:
Anders
->
Madison
Prev:
0000438d7625b86a6f366545b1929975a0d3ff1f8847e56cc587cadddb0ab781
Hash:
0000baeab68c2a60f9a6fa56355438d97c672a15494fcea617064d9314f9ff63
Mine

What is a blockchain?

…a **consensus-supported**, **ledger** of **cryptographically signed** statements that are recorded in an **agreed-upon order** (in a **deterministic** way). ARGH!

5 Key Developments in Blockchain

- Emergence of Bitcoin
- Separation from Bitcoin
- Smart Contracts
- Proof of Stake
- Scaling blockchain tech

Source: “A brief history of Blockchain Technology,” by Bernard Marr, Forbes. (16 Feb 2018)

Emergence of Bitcoin (Dev1)

- Bitcoin: A Peer to Peer Electronic Cash System by Satoshi Nakamoto
  - Describes a purely decentralized currency (2008)
  - Potentially transformative and disruptive technology
  - Release of Bitcoin to open source community (2009)
- Cryptocurrency is the first “killer app” for blockchain tech
  - Ecosystem begins to explode (miners, traders, merchants, consumers…)
- “Blockchain is to Bitcoin, what the Internet is to email. A big electronic system, on top of which you build applications.”
  - Sally Davies, FT Technology Reporter
- Blockchain ≠ Bitcoin

Separation from Bitcoin (Dev2)

- Blockchain can be used beyond cryptocurrency
- A decentralized ledger permanently recording transactions between parties
- No need for third party authentication (Efficiency! Lower cost!)
- People start to ask, how can blockchain help me?
- Supply chains, healthcare, insurance, transportation, voting, contracts, …

What is blockchain?

- Hashing
- Block
- Block chain
- Distributed block chain
- …and a dash of crypto

What is hashing?

- A hash is a fingerprint of some input data.
  - Can be used to retrieve data _and_ verify that data hasn’t changed.

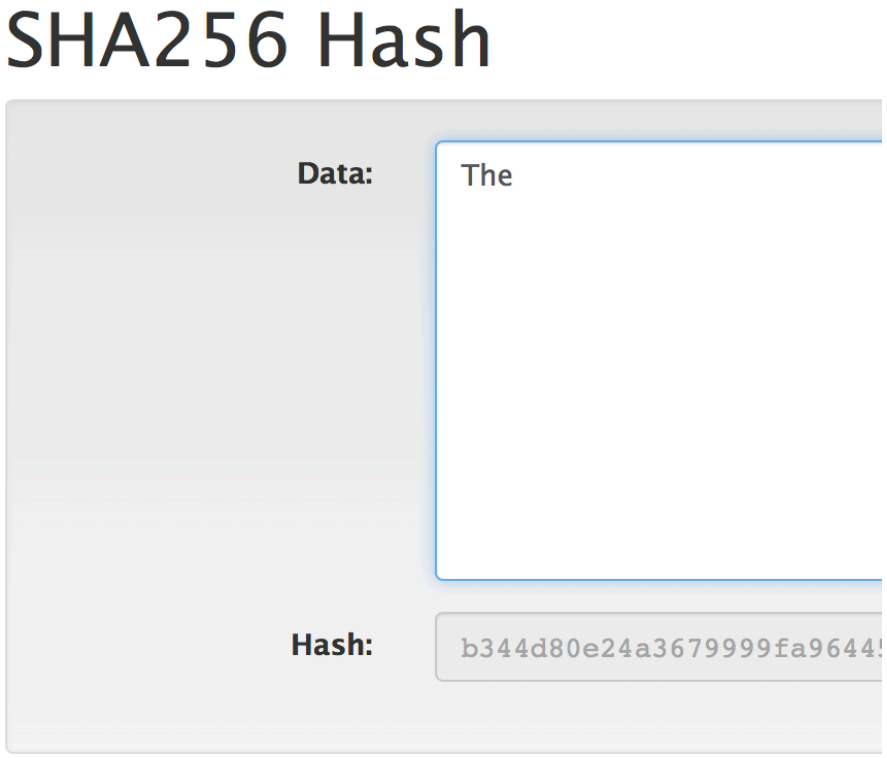
- A hash function is any function that can be used to map data of arbitrary size to data of a fixed size.

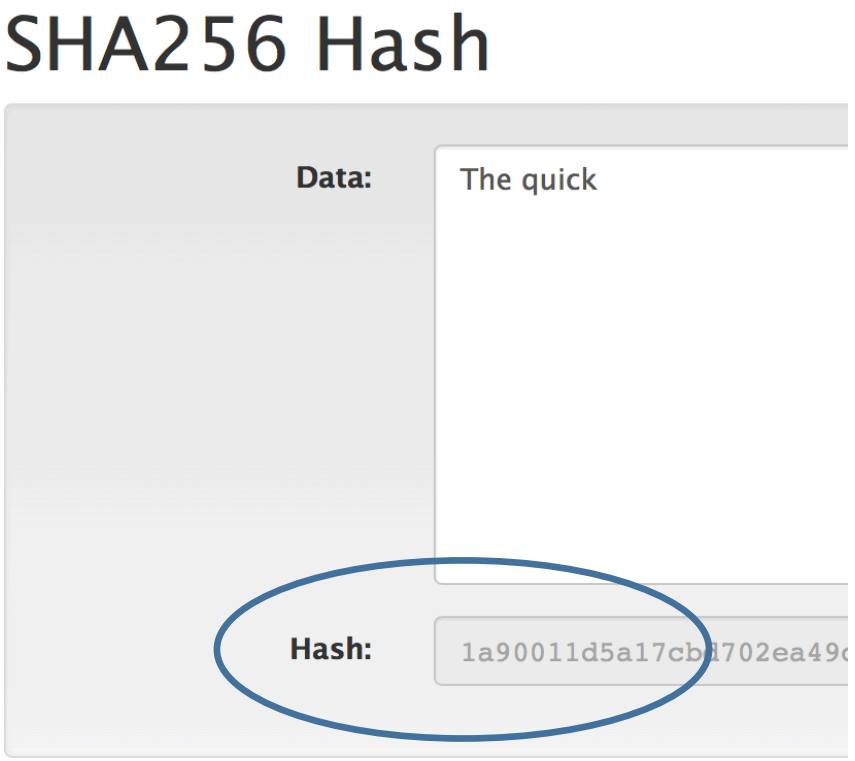
- SHA = Secure Hashing Algorithm
  - SHA256 takes arbitrary data maps to 256 bits (displays as HEX)
  - Produces irreversible and unique hashes (used in crypto)
  - Compare resulting hash of data rather than comparing original data

Want more? Crypto courses: CS 4264, MATH 4175 + 4176, ECE 5580

256-bit hashing

Digital fingerprint of data. Deterministic. Many to 256 mapping.





Source: https://anders.com/blockchain/ See also: https://www.gizillion.org by Scott Stornetta

What is a block?

- Elements of a block
  - Block number: 1
  - Nonce: 72608
  - Data: <empty>
  - Using SHA256 results in:
    - HASH: xxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxx
- What if we know the block number and the data is fixed.
  - And we know we want HASH to have 4 leading zeros.
  - HASH: 0000xxxxxxxxxxxx xxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxx
  - Can we find the Nonce to complete the block? YES!
  - [i.e., can we ALTER the data and still produce a valid HASH?]

Here, x is a hexadecimal value from 1 to f.

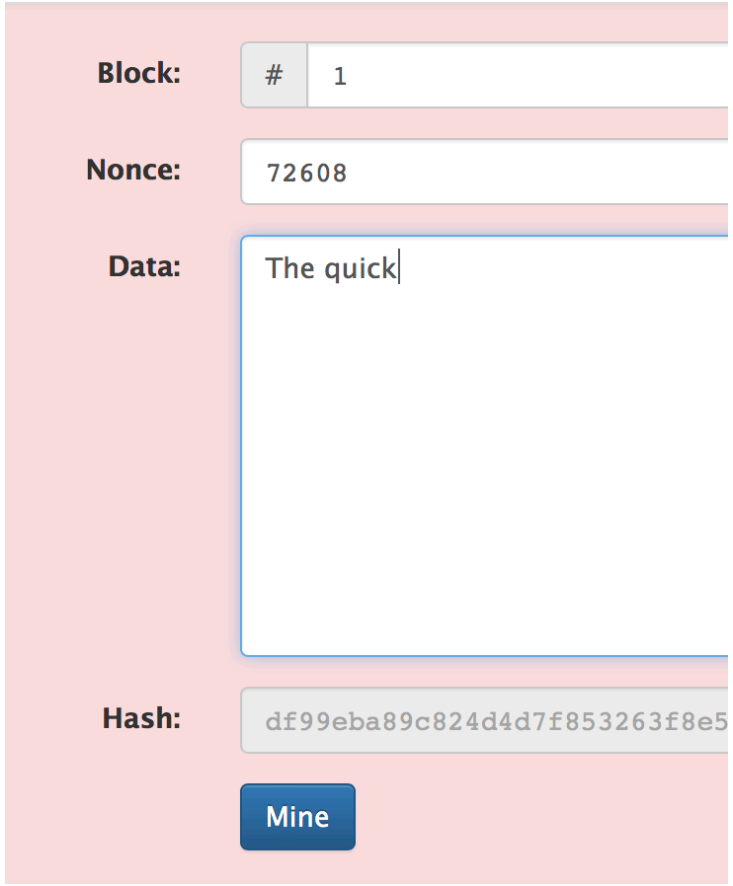
What is the Nonce?

- NONCE = Number used only once
- Nonce is number added to a hashed block that when rehashed meets difficulty level restrictions of the TARGET HASH.
- Nonce is the number that blockchain miners are searching for.

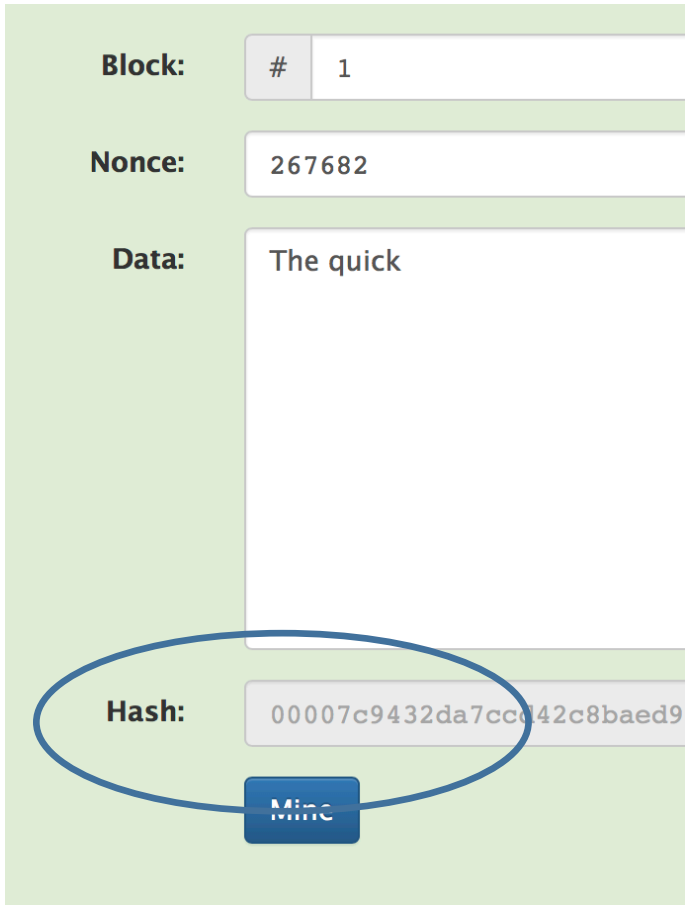
- TARGET HASH = Number that a hashed block header must be less than or equal to in order for a new block to be awarded. Can be adjusted to match desired difficulty level (leading zeros).

Mining the Nonce

Extend hash idea to block. #+Nonce+data➔hash.



Modify any of (#+Nonce+data) invalidates (breaks) block.

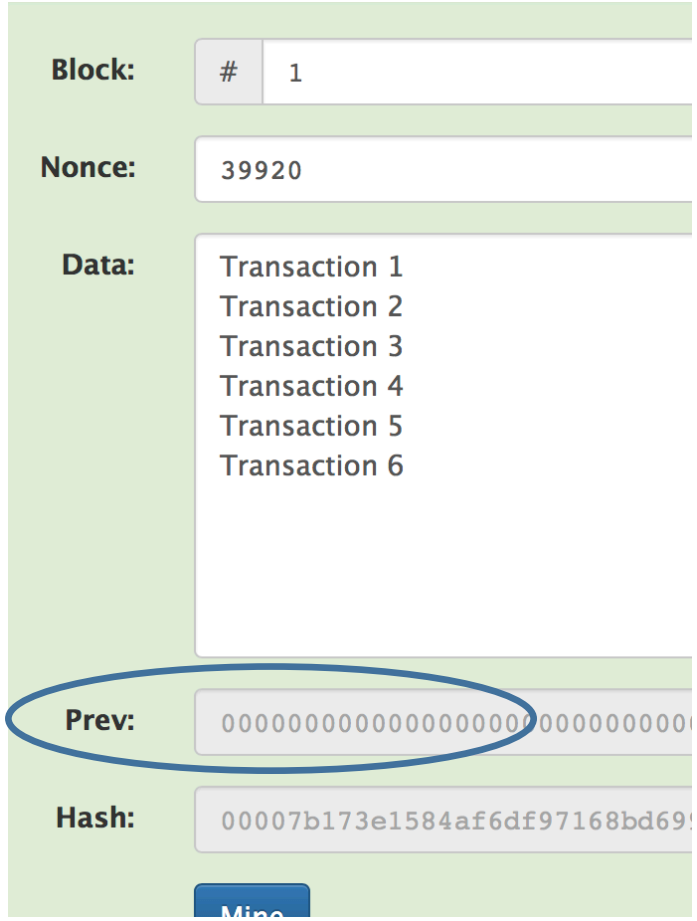


Target hash requires 4 zeros. Zeros➔difficulty.

Takes a lot of “work” to MINE the Nonce.

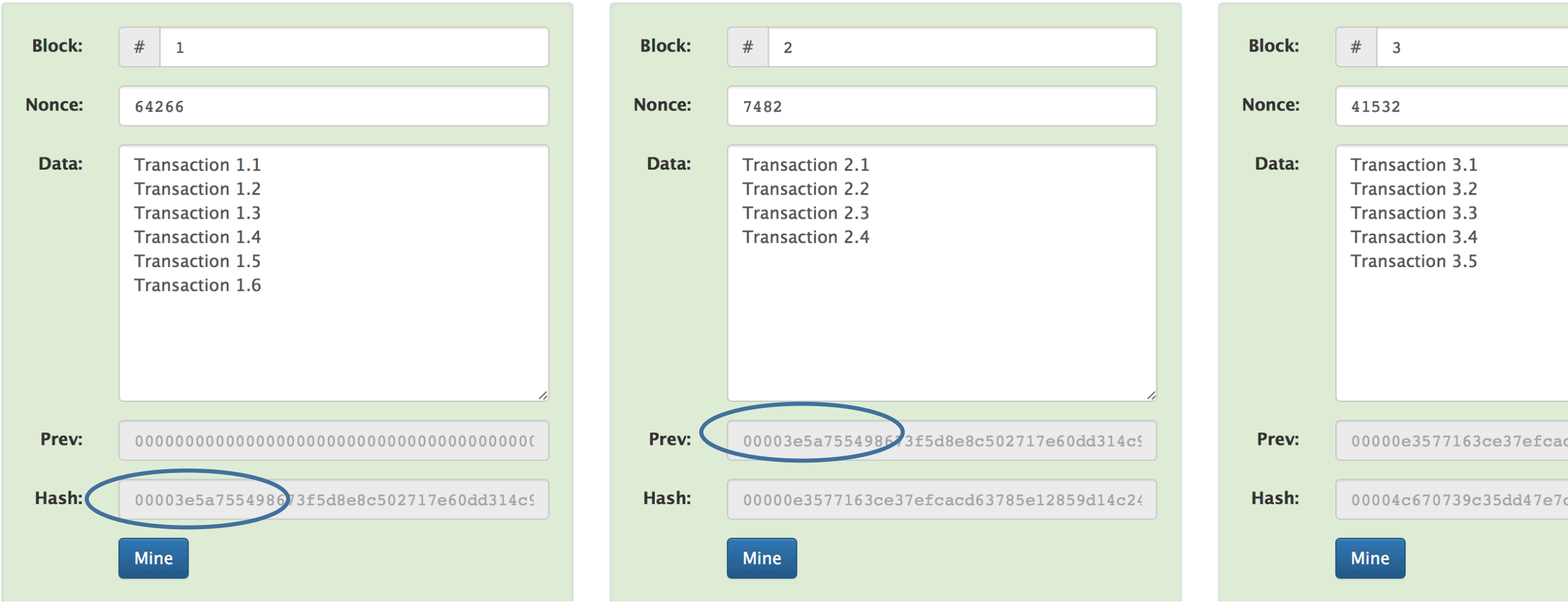
What is a block in a blockchain?

- A block is a group of transactions in **chronological** order
- Each block is made of a Block Header and a “Block Body.”
- Every block has, as its data, the hash of the previous block.
- Note: In our examples, a “signed block” has 4 leading zeros.



First block in chain is called “genesis” block. No previous block.

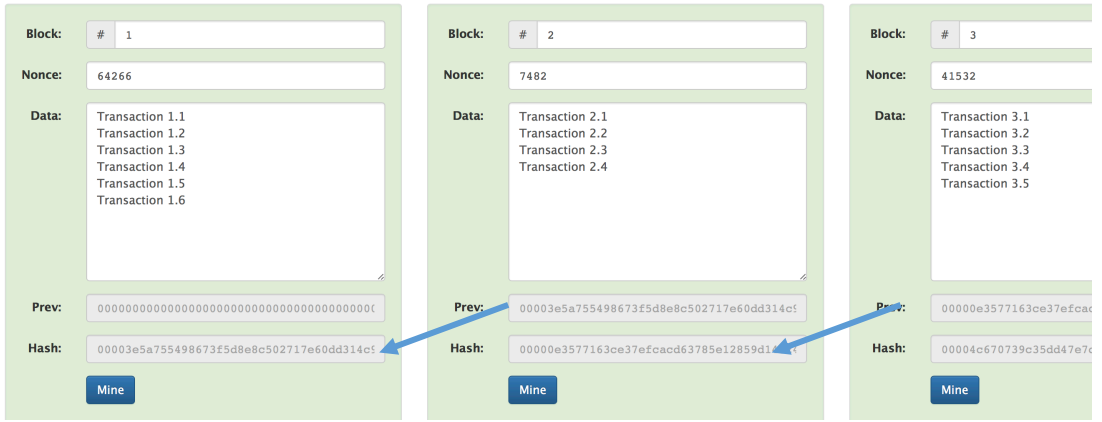
What is a blockchain?



Can I modify the transaction records in this blockchain? YES, but…

What is a blockchain?

- Each block knows hash of previous block.
- A linked list of blocks using hash pointers.
- Signed hash here has 4 leading zeros.



What happens if I change last block?

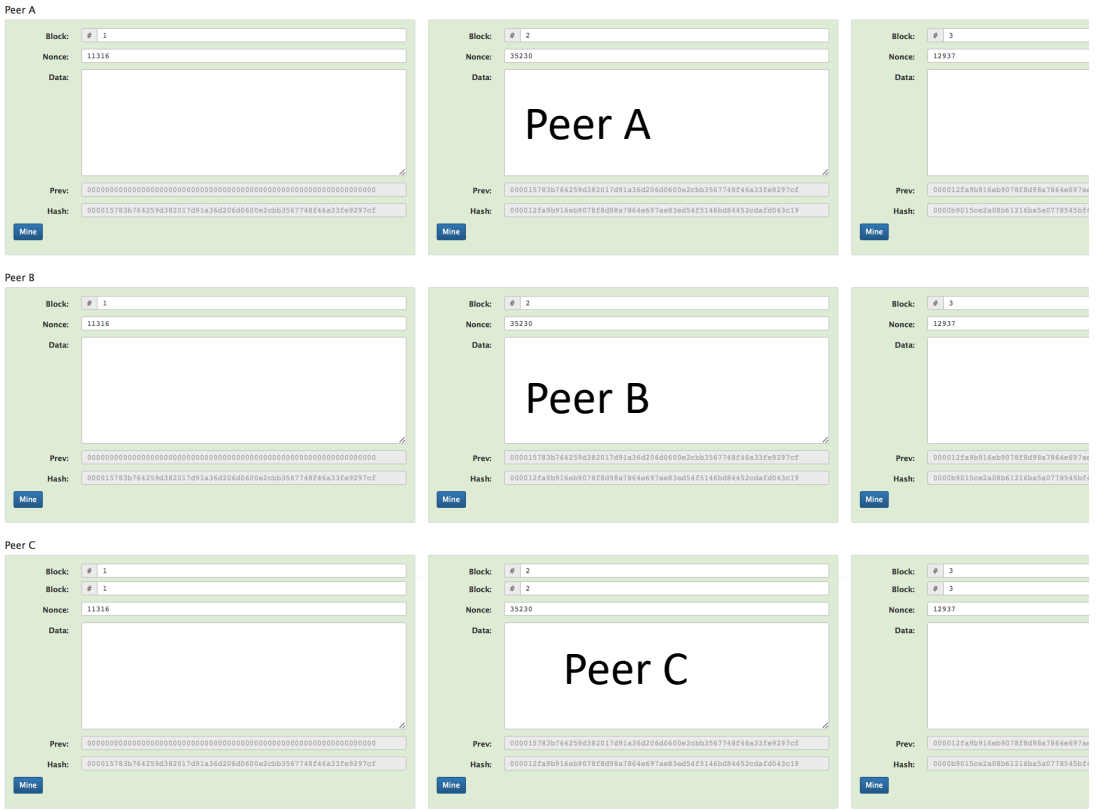
What happens if I change an earlier block?

What if I re-mine the last block? Earlier block?

What does it mean to be immutable?

Distributed Blockchain

- How to alter the previous chain?
  - Modify the transaction data in the block header
  - Re-mine the Nonce to get a hash with 4 zeros
  - Do the same for each succeeding block (sequentially)
- Distributed Blockchain
  - Each PEER has a complete copy of the blockchain



Distributed Blockchain

- How to alter a single copy of the chain?
  - Modify the transaction data in the block header
  - Re-mine the Nonce to get a hash with 4 zeros
  - Do the same for each succeeding block (sequentially)
- Distributed Blockchain
  - Each PEER has a complete copy of the blockchain
- Example:
  - Peers A thru C have exact copy of blockchain
  - Peer B changes DATA and re-mines Nonce to get valid 4-leading-zeros hash
  - However, Peer B's hashes don't match Peers A and C's hashes. Simple check.

Peer A

Block:	# 1	# 2	# 3
Nonce:	64266	78974	38598
Data:	Transaction 1.1 Transaction 1.2 Transaction 1.3 Transaction 1.4 Transaction 1.5 Transaction 1.6	Transaction 2.1 Transaction 2.2 Transaction 2.3 Transaction 2.4 Transaction 2.5 Transaction 2.6	Transaction 3.1 Transaction 3.2 Transaction 3.3 Transaction 3.4 Transaction 3.5 Transaction 3.6
Prev:	0000000000000000000000000000000000000000000000000000000000000000	00003e5a755498673f5d8e8c502717e60dd314c90dfa7c4130dbf1983a5830ce	0000fc21fe83e19978498c68808997b70b6f8eb229
Hash:	00003e5a755498673f5d8e8c502717e60dd314c90dfa7c4130dbf1983a5830ce	0000fc21fe83e19978498c68808997b70b6f8eb2294ff9aac502427de7a6370c	0000827c3aee25ac0747f70544a5b94f2cabc37db8
	Mine	Mine	Mine

Peer B

Block:	# 1	# 2	# 3
Nonce:	64266	78974	38598
Data:	Transaction 1.1 Transaction 1.2 Transaction 1.3 Transaction 1.4 Transaction 1.5 Transaction 1.6	Transaction 2.1 Transaction 2.2 Transaction 2.3 Transaction 2.4 Transaction 2.5 Transaction 2.6	Transaction 3.1 Transaction 3.2 Transaction 3.3 Transaction 3.4 Transaction 3.5 Transaction 3.6
Prev:	0000000000000000000000000000000000000000000000000000000000000000	00003e5a755498673f5d8e8c502717e60dd314c90dfa7c4130dbf1983a5830ce	0000fc21fe83e19978498c68808997b70b6f8eb229
Hash:	00003e5a755498673f5d8e8c502717e60dd314c90dfa7c4130dbf1983a5830ce	0000fc21fe83e19978498c68808997b70b6f8eb2294ff9aac502427de7a6370c	0000827c3aee25ac0747f70544a5b94f2cabc37db8
	Mine	Mine	Mine

Peer C

Block:	# 1	# 2	# 3
Nonce:	64266	78974	38598

What happens if I change last block?

What happens if I change an earlier block?

What if I re-mine the last block? Earlier block?

What does it mean to be immutable?

How long does it take to find a mutation?

Peer A

Block: # 1
Nonce: 64266
Data:
Transaction 1.1
Transaction 1.2
Transaction 1.3
Transaction 1.4
Transaction 1.5
Transaction 1.6
Prev: 00000000000000000000000000000000000000000000000000000000000000000000
Hash: 00003e5a755498673f5d8e8c502717e60dd314c90dfa7c4130dbf1983a5830ce
Mine

Block: # 2
Nonce: 78974
Data:
Transaction 2.1
Transaction 2.2
Transaction 2.3
Transaction 2.4
Transaction 2.5
Transaction 2.6
Prev: 00003e5a755498673f5d8e8c502717e60dd314c90dfa7c4130dbf1983a5830ce
Hash: 0000fc21fe83e19978498c68808997b70b6f8eb2294ff9aac502427de7a6370c
Mine

Block: # 3
Nonce: 38598
Data:
Transaction 3.1
Transaction 3.2
Transaction 3.3
Transaction 3.4
Transaction 3.5
Transaction 3.6
Prev: 0000fc21fe83e19978498c68808997b70b6f8eb229
Hash: 0000827c3aee25ac0747f70544a5b94f2cabc37db8
Mine

Peer B

Block: # 1
Nonce: 64266
Data:
Transaction 1.1
Transaction 1.2
Transaction 1.3
Transaction 1.4
Transaction 1.5
Transaction 1.6
Prev: 00000000000000000000000000000000000000000000000000000000000000000000
Hash: 00003e5a755498673f5d8e8c502717e60dd314c90dfa7c4130dbf1983a5830ce
Mine

Block: # 2
Nonce: 78974
Data:
Transaction 2.1
Transaction 2.3
Transaction 2.x
Transaction 2.5
Prev: 00003e5a755498673f5d8e8c502717e60dd314c90dfa7c4130dbf1983a5830ce
Hash: 13594a3e52f2b22a6ef4028b8a9d87f054ed2cb55c844508cac2cdf4709b045f
Mine

Immutable

Block: # 3
Nonce: 38598
Data:
Transaction 3.1
Transaction 3.2
Transaction 3.3
Transaction 3.4
Transaction 3.5
Transaction 3.6
Prev: 13594a3e52f2b22a6ef4028b8a9d87f054ed2cb55c
Hash: 8d959394e79ebb84782dd4222f653d2c31b953c18d
Mine

Peer C

Block: # 1

Block: # 2

Block: # 3

Peer A

Block:	# 1	# 2	# 3
Nonce:	64266	78974	38598
Data:	Transaction 1.1 Transaction 1.2 Transaction 1.3 Transaction 1.4 Transaction 1.5 Transaction 1.6	Transaction 2.1 Transaction 2.2 Transaction 2.3 Transaction 2.4 Transaction 2.5 Transaction 2.6	Transaction 3.1 Transaction 3.2 Transaction 3.3 Transaction 3.4 Transaction 3.5 Transaction 3.6
Prev:	0000000000000000000000000000000000000000000000000000000000000000	00003e5a755498673f5d8e8c502717e60dd314c90dfa7c4130dbf1983a5830ce	0000fc21fe83e19978498c68808997b70b6f
Hash:	00003e5a755498673f5d8e8c502717e60dd314c90dfa7c4130dbf1983a5830ce	0000fc21fe83e19978498c68808997b70b6f8eb2294ff9aac502427de7a6370c	0000827c3aee25ac0747f70544a5b94f2cab
	Mine	Mine	Mine

Peer B

Block:	# 1	# 2	# 3
Nonce:	64266	2142	55801
Data:	Transaction 1.1 Transaction 1.2 Transaction 1.3 Transaction 1.4 Transaction 1.5 Transaction 1.6	Transaction 2.1 Transaction 2.2 Transaction 2.3 Transaction 2.x Transaction 2.5 Transaction 2.6	Transaction 3.1 Transaction 3.2 Transaction 3.3 Transaction 3.4 Transaction 3.5 Transaction 3.6
Prev:	0000000000000000000000000000000000000000000000000000000000000000	00003e5a755498673f5d8e8c502717e60dd314c90dfa7c4130dbf1983a5830ce	0000b53f2f3df537ab1a341f4c22f481343f
Hash:	00003e5a755498673f5d8e8c502717e60dd314c90dfa7c4130dbf1983a5830ce	0000b53f2f3df537ab1a341f4c22f481343f1f02e5d8b1230e8839e19f23d30c	0000c3bc6b912a9e6e8bc536fa854e5efc12
	Mine	Mine	Mine

Peer C

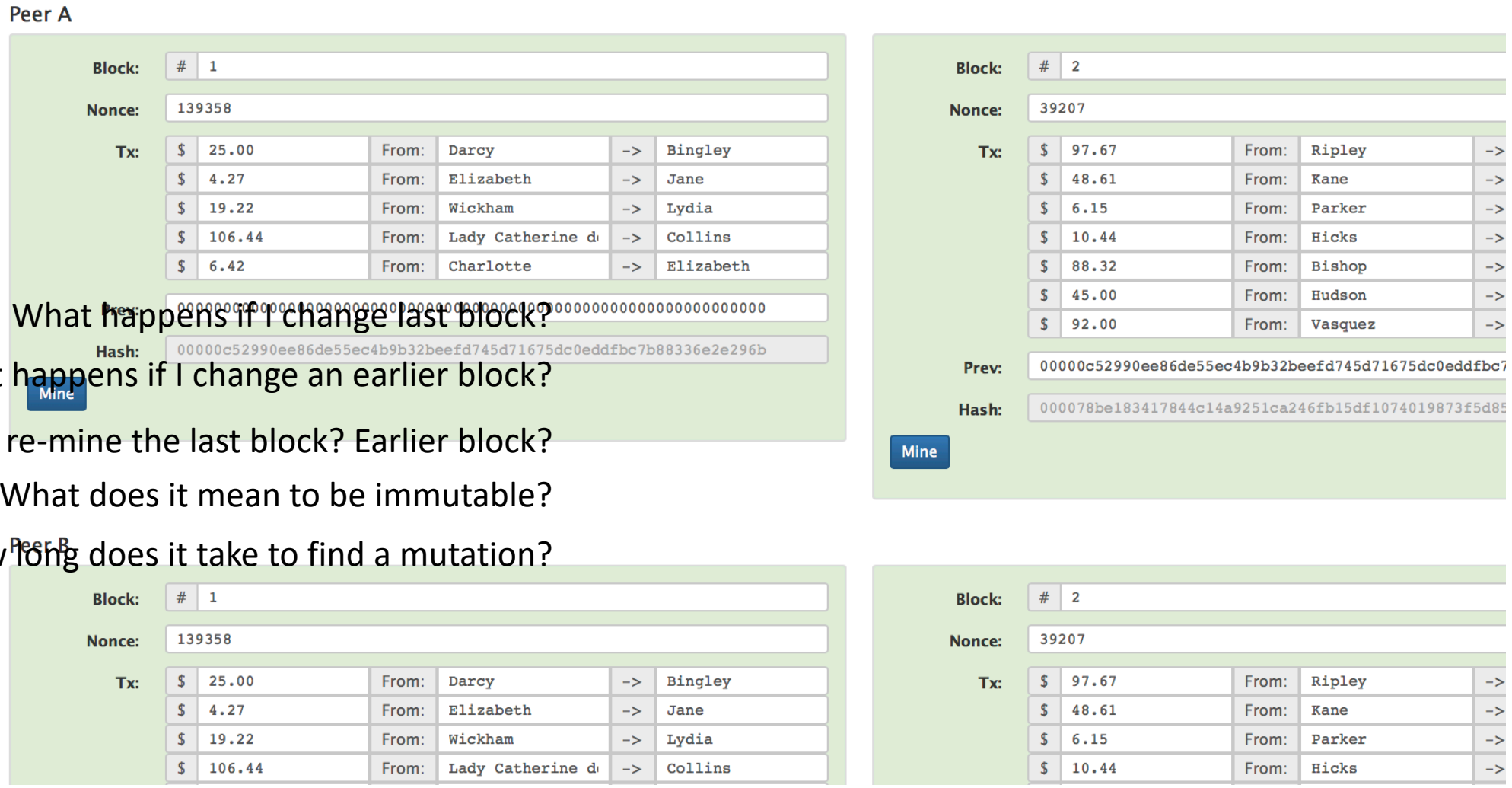
Block:	# 1	# 2	# 3
Nonce:	64266	78974	38598

What happens next? Consensus.

- Peers have different “correct” copies of the blockchain.
- But, only one copy has been modified (e.g., change transaction)
- All other peers with copies can vote on which version is correct
- If 51% or more agree on a change to the blockchain, all must update
  - Or be left out, unable to participate until agreeing on mods
- Two key CONSENSUS algorithms we will cover
  - Proof of Work
  - Proof of Stake

Token ledger example

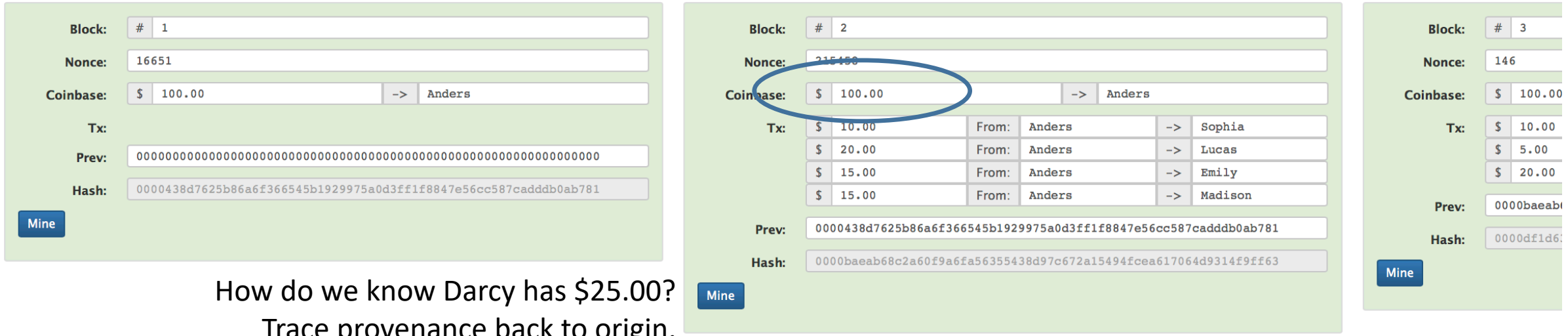
What kind of data does the blockchain track? Example: ledger.



How do we know Darcy has $25.00?

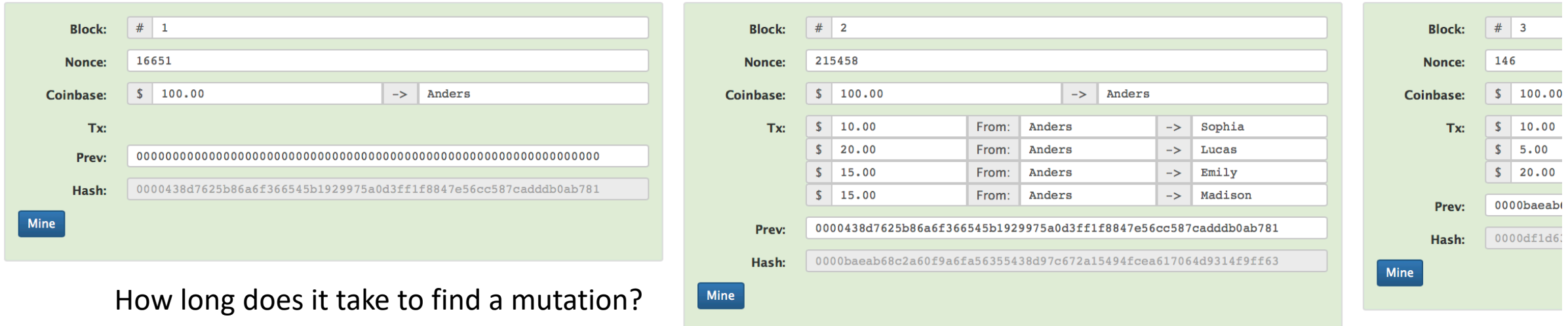
Transaction ledger example

Peer A



How do we know Darcy has $25.00?
Trace provenance back to origin.
Immutable history of transactions.

Peer B



How long does it take to find a mutation?

Proof of Work

- Requires users to perform some form of work to participate
- Must be difficult to produce, easy to verify (asymmetry)
- Example: Bitcoin miner nodes compete to solve a puzzle (block)
  - First correct solution awarded with a batch of coins
  - Earlier example, guess the nonce correctly, fastest
  - Difficulty increases proportional to total compute power on network
  - Makes for a competitive system with rewards; optimizes verify step
  - 51% of compute resources affect acceptance/denial of change in blockchain

Think carrots.

Smart Contracts (Dev3)

- Next Killer App for Blockchain
- Exchange money, property or services without centralized authority
- If-then clauses tied to exchange
- Real estate transaction: You place deposit, I send key by date.
  - If no key by date, refund you. If key by date, deposit to me, key to you.
  - Blockchain community “judges” the contract validity through consensus.
  - Transaction recorded permanently on the blockchain.

Consensus Alternatives?

- Proof of Work Merits
  - Widely used
  - Robust, working solution for a decade
  - “Harder” to hack with scale
- Proof of Work Limitations
  - Mining is wasteful
  - Tragedy of the commons (when bounties disappear, incentives decrease)
  - Smaller network (or oligarchy via collusion) can lead to manipulation

Proof of Stake (Dev4)

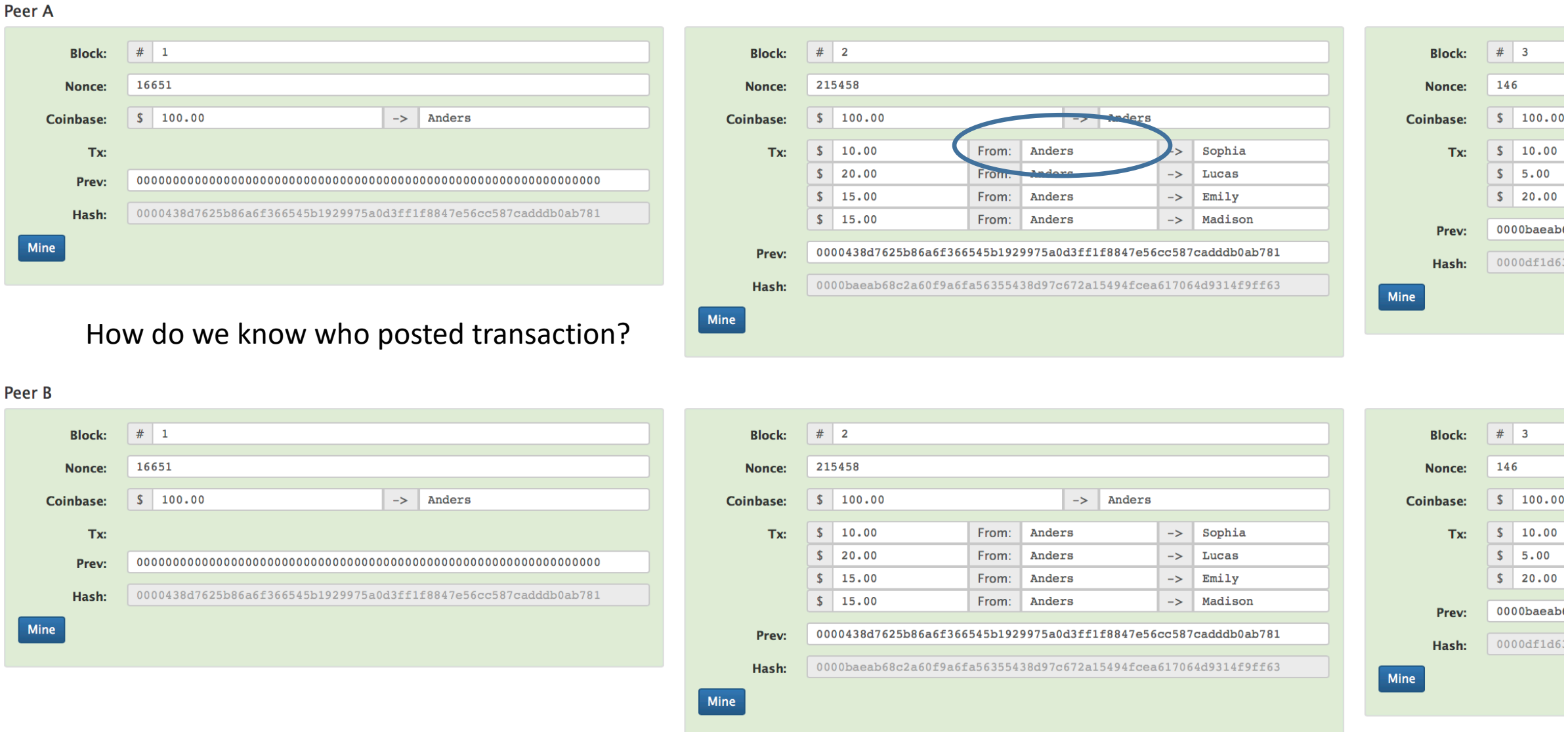
- Serves same purpose as Proof of Work
  - Adding to blockchain and verification of new blocks by p2p network
- Creator of new blocks deterministic (linked to wealth or stake)
  - Chosen from pool of users with stake
- “Miners” take transaction fee for producing blocks
- Penalty is possible loss of stake for bad actors
- 51% threat based on “supply” rather than computing power

Think sticks.

Proof of Stake (Dev4)

- Merits
  - Greener and cheaper form of consensus than Proof of Work
  - Seen as promising alternative to Proof of work
  - “Harder” to hack at scale
  - Can be faster to reach consensus
- Limitations
  - Not as widespread adoption as Proof of Work (but growing)
  - Can potentially be manipulated with collusion
- Extension
  - Delegated proof of stake (DPOS)

Transaction ledger example



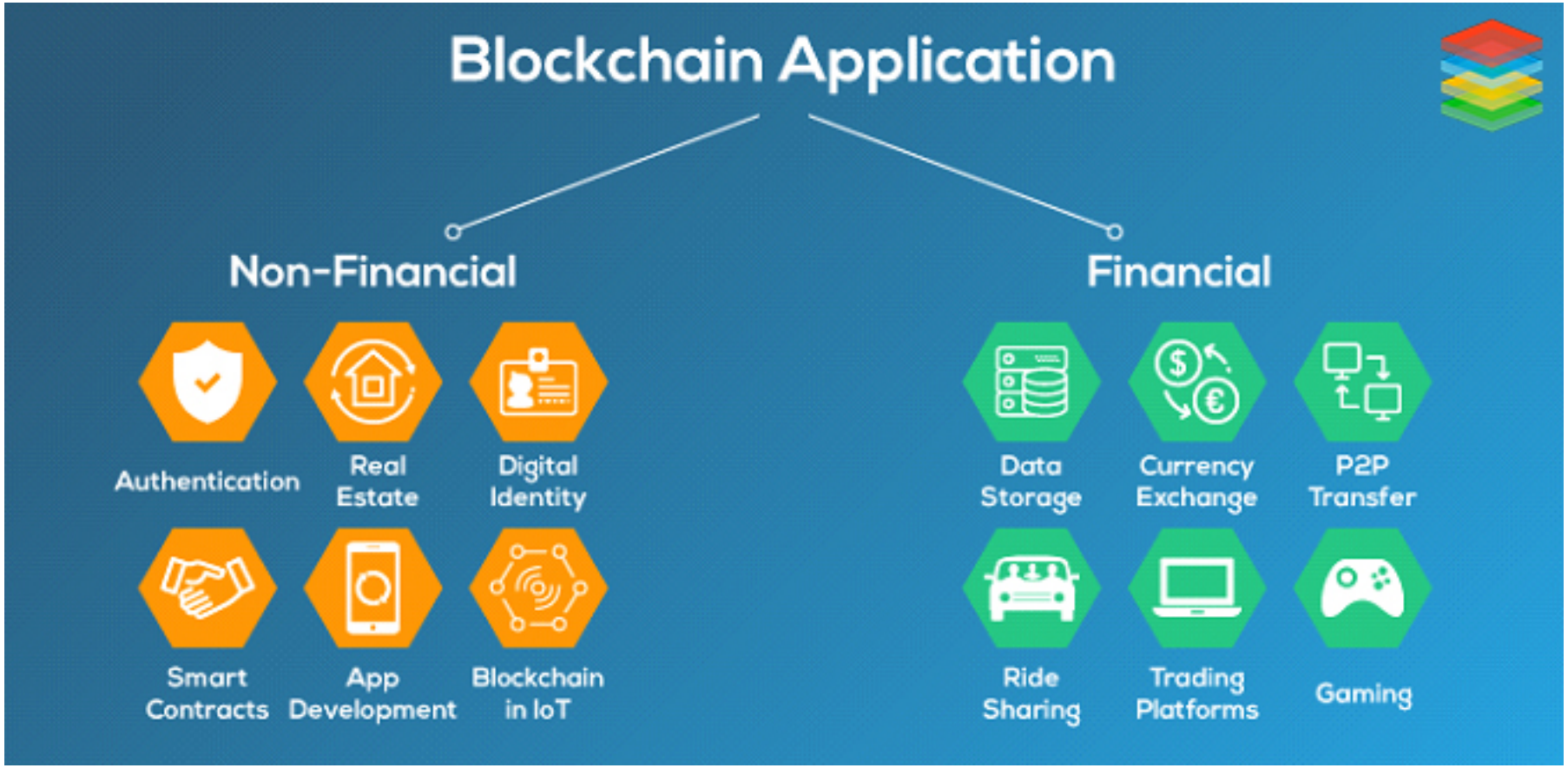
What is Blockchain?

> …a **consensus-supported**, **ledger** of **cryptographically signed** statements that are recorded in an **agreed-upon order** (in a **deterministic** way). ARGH!

- Ledger: Blocks of transaction history (deterministic order)
- Consensus: Proof of Work/Stake
- Cryptographically signed: Public/Private Key Pairs

What's Next?

- Time to start making some blockchain apps!



A Gentleman's Guide to Crypto

- Cryptography: Process of communicating securely in an insecure environment.
  - Convert message you want to send to a CIPHER text (gibberish without secret to unlock)

- Symmetric cryptography
  - Implies sender and receiver are in control of the cipher
  - Example: Julius Caesar shifted message sent by # letters in alphabet.
  - (encrypt: shift right by 1) Kirk Cameron --> ljsl dbnfspo (decrypt shift left by 1)
  - Challenges: 1) Can be easy to decrypt; 2) How to tell others the cipher.

Public Key Cryptography

- One asymmetric solution
- RECEIVER: Generates a KEY PAIR (Public + Private)
  - Sends Public Key to SENDER
- SENDER: Encrypt message using RECEIVER's Public Key
  - Sends encrypted message to RECEIVER
- RECEIVER: Decrypts encrypted message using private key

- Private keys cannot be derived from public keys.

Generate public-private key (RSA-like)

- PICK two prime numbers: P=3, Q=11
  CALCULATE N=P*Q=3x11=33
  CALCULATE Z=(P-1)*(Q-1)=(3-1)*(11-1)=20
- CHOOSE a prime number K such that K is co-prime to Z
  (prime K is coprime to Z if Z is not divisible by K and K is prime)
  Pick K=7 (from list of 3, 7, 11, 13, 17, 19)
  N=33 and K=7 are Server's PUBLIC KEY
- SOLVE (K*J) MOD Z = 1 for J
  (7*J) MOD 20 = (21) MOD 20 = 1; if J=3
  N=33 and J=3 are Server's PRIVATE KEY

ECRYPT, SEND, DECRYPT

- ENCRYPT + DECRYPT

ENCRYPT "14”

Use Public Key (K,N)=(7,33) to encrypt

<encoded messg>^K MOD N = 14^7 MOD 33 ==> 20 --> encrypted messg

Send encrypted transmission ”20”

DECRYPT “20”

Use Private Key (J,N)=(3,33) to decrypt

<encrypted messg>^J MOD N = 20^3 MOD 33 ==> 14 --> decrypted messg

Generate public-private key (RSA-like)

- PICK two prime numbers: P=2, Q=7
  CALCULATE N=P*Q=2*7=14
  CALCULATE Z=(P-1)*(Q-1)=(2-1)*(7-1)=6
- CHOOSE a prime number K such that K is co-prime to Z
  (prime K is coprime to Z if Z is not divisible by K and K is prime)
  Pick K=5 (from list of 3,5)
  N=14 and K=5 are PUBLIC KEY
- SOLVE (K*J) MOD Z = 1 for J
  (5*J) MOD 6 = 25 MOD 6 = 1; if J=11
  N=14 and J=11 are PRIVATE KEY (J=5 works, but then PUBLIC=PRIVATE)
  [J=5,11,17,23,29,35,41,47 all work if J<50]

ECRYPT, SEND, DECRYPT

- ENCRYPT + DECRYPT

ENCRYPT "2”

Use Public Key (K,N)=(5,14) to encrypt

<encoded messg>^K MOD N = 2^5 MOD 14 ==> 4 --> encrypted messg

Send encrypted transmission "4”

DECRYPT "4”

Use Private Key (J,N)=(11,14) to decrypt

<encrypted messg>^J MOD N = 4^11 MOD 14 ==> 2 --> decrypted messg

Normal message exchange

- Alice wants to echange a key with Bob.
- Alice has a pair of keys (kpubA and kprivA)
- Alice sends her public key kpubA to Bob
- Bob picks random key (k), uses F(kpubA,k) to encrypt and send to Alice
- Alice decrypts F(kpubA,k) using her private key kprivA to find k

Man-in-the-middle attack

- MITM Monitors all transmissions between Alice and Bob
- MITM has a pair of keys (kpubM and kprivM)
- MITM intercepts initial message from Alice to Bob containing public key kpubA
- MITM sends public key kpubM to Bob (pretending to be Alice)
- Bob picks random key (k), uses F(kpubM,k) to encrypt and send to MITM (fake Alice)
- MITM decrypts F(kpubM,k) using private key kprivM to find k
- MITM uses random key (k), creates F(kpubA,k) to encrypt and send to Alice
- Alice thinks her and Bob are communicating securely with common key k known only to them
- MITM can now decrypt all messages between Alice and Bob

Stopping MITM Attack

- Trusted 3rd party confirms (for Bob) he has received Alice's public key (e.g., TLS)
- Secure cite anyone can access to READ ONLY anyone else's public key

    Bob can simply compare kpubM to Alice's public key to reveal MITM

    OR Alice can generate key (k) herself, and double encrypt with Bob's public key

    Bob uses his private key to decrypt outer encryption F(kpubB,F(kprivA,k))

    Bob then uses Alice's public key to decrypt inner encryption to find k

See the connection to Blockchain yet?

Cryptographically Signed Transactions

Applying YOUR private key to encryption is called "Signing" or a "Digital Signature”

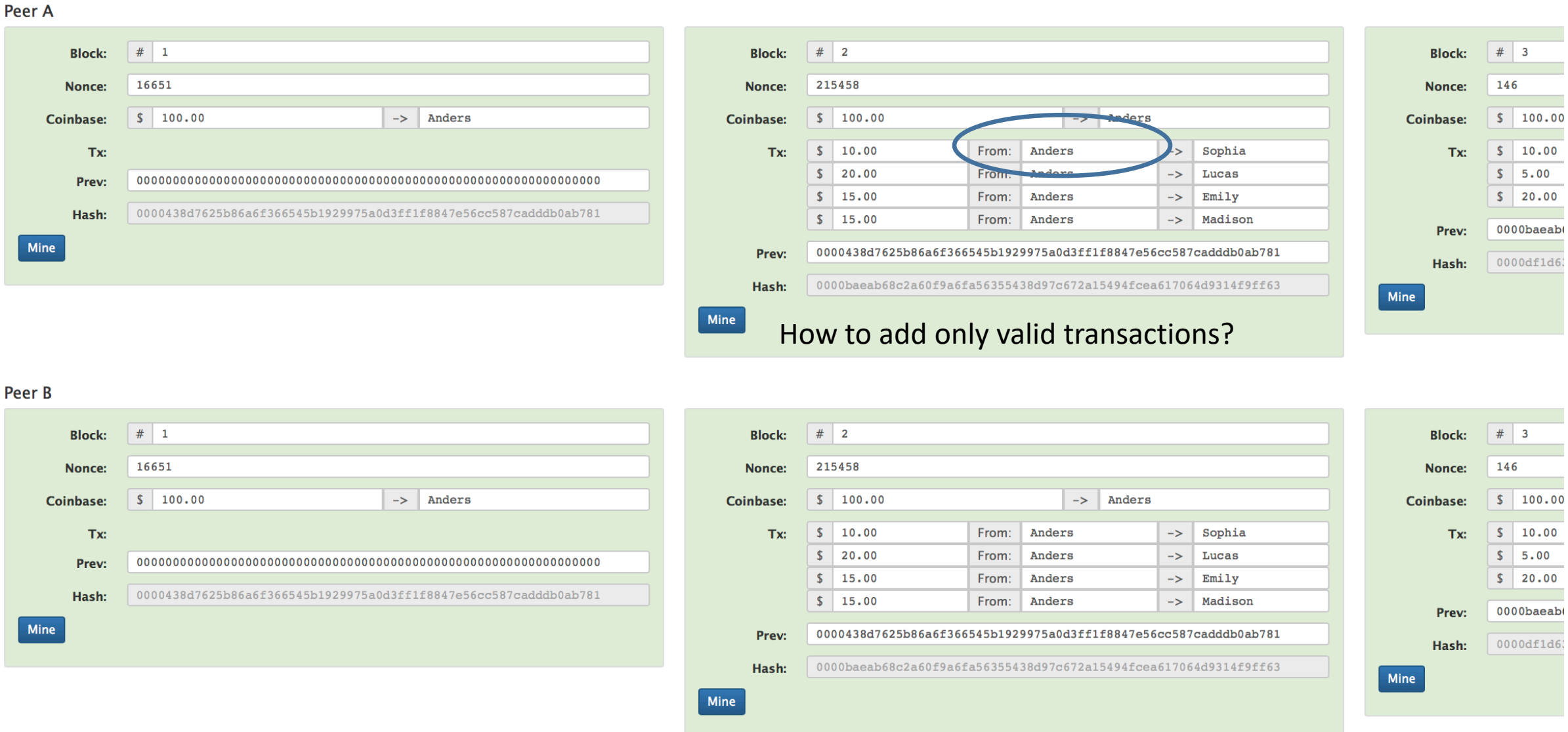
Anything encrypted by YOUR PUBLIC key can only be opened by YOUR private key

Keeps data secured and only viewable by you (or using your private key)

Anything encrypted by YOUR PRIVATE key can only be opened by YOUR public key

Thus proving you locked it in the first place

Transaction ledger example

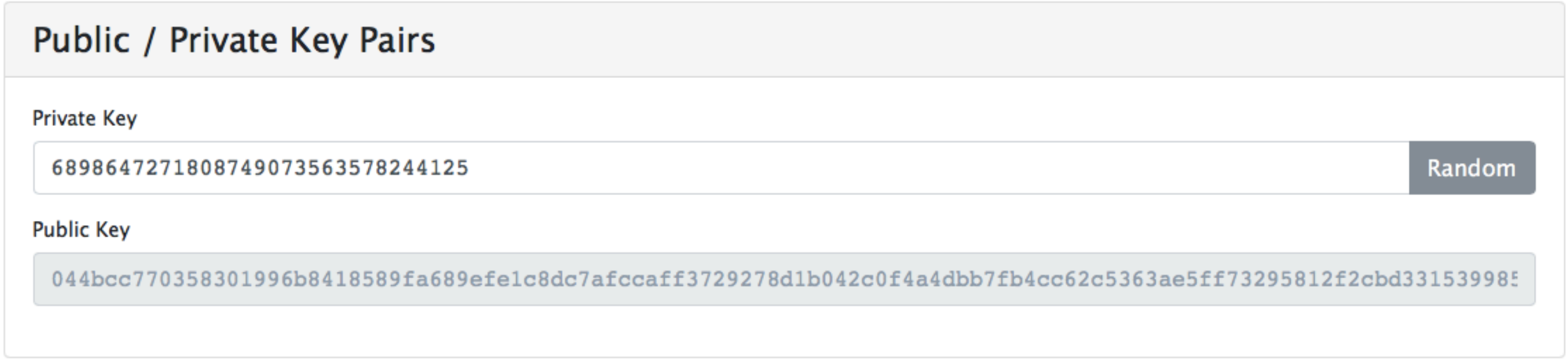


Public-Private Key Pairs

Creation is fast and cheap. Anonymity.

Electronic signature/verification (Gen Pair)

Step 1 for signing a transaction.



Encrypt Message with Private Key

Message + private key → message sig (signed)

Let's meet at the skate park.

Private Key

68986472718087490735635782441 25

Sign

Message Signature

304402204eb5494b8d64e2b868e2e058a23b2925bc2453e53d00262e5d4e1c30bb532f7202200acf09de65d5f7e00dcc0fe6bbe71aa5(

Decrypt with Public Key (Verify Sender)

Whoever signed this had access to private key of this public key holder.

Message

Let's meet at the skate park.

Public Key

044bcc770358301996b8418589fa689efe1c8dc7afccaff3729278d1b042c0f4a4dbb7fb4cc62c5363ae5ff73295812f2cbd33153998

Validity of transaction checked with public key.

Signature

304402204eb5494b8d64e2b868e2e058a23b2925bc2453e53d00262e5d4e1c30bb532f7202200acf09de65d5f7e00dcc0fe6bbe71aa5(

Verify

Signature improperly modified

Message

Let's meet at the skate park.

Public Key

044bcc770358301996b8418589fa689efe1c8dc7afccaff3729278d1b042c0f4a4dbb7fb4cc62c5363ae5ff73295812f2cbd33153998

Validity of transaction checked with public key.

Signature

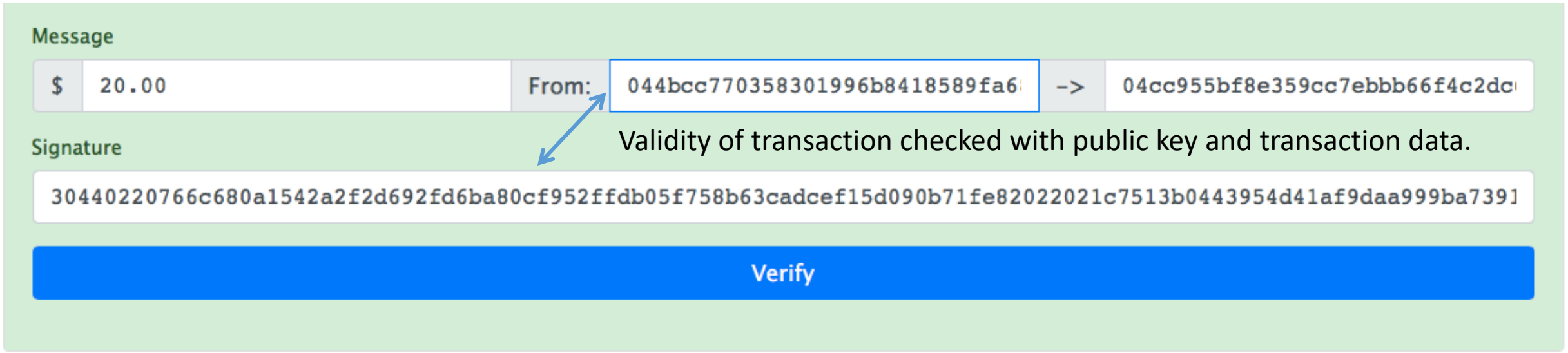
304402204b5494b8d64e2b868e2e058a23b2925bc2453e53d00262e5d4e1c30bb532f7202200acf09de65d5f7e00dcc0fe6bbe71aa501

Verify

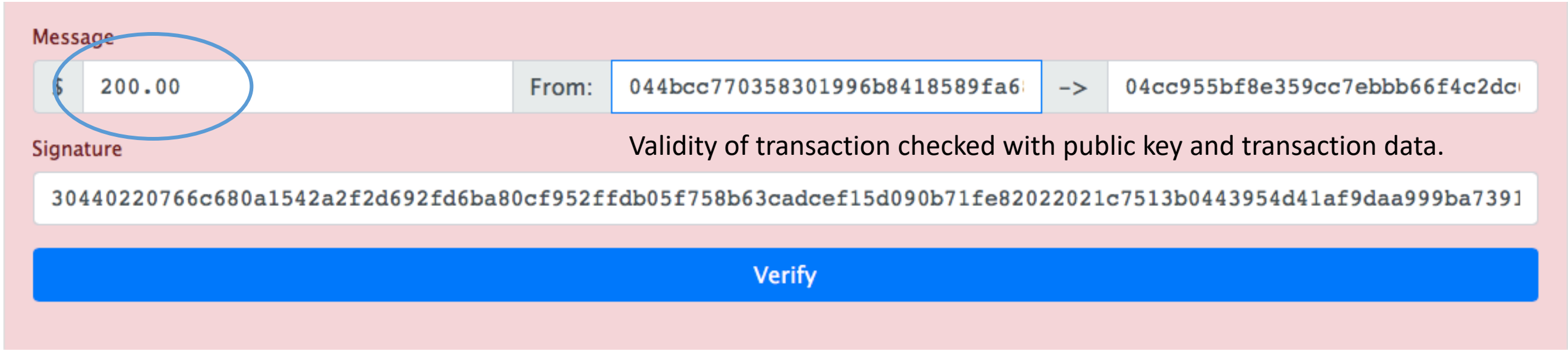
Encrypting Transactions (w/ private)

Verifying with public key

Whoever signed this had access to private key of this public key holder.

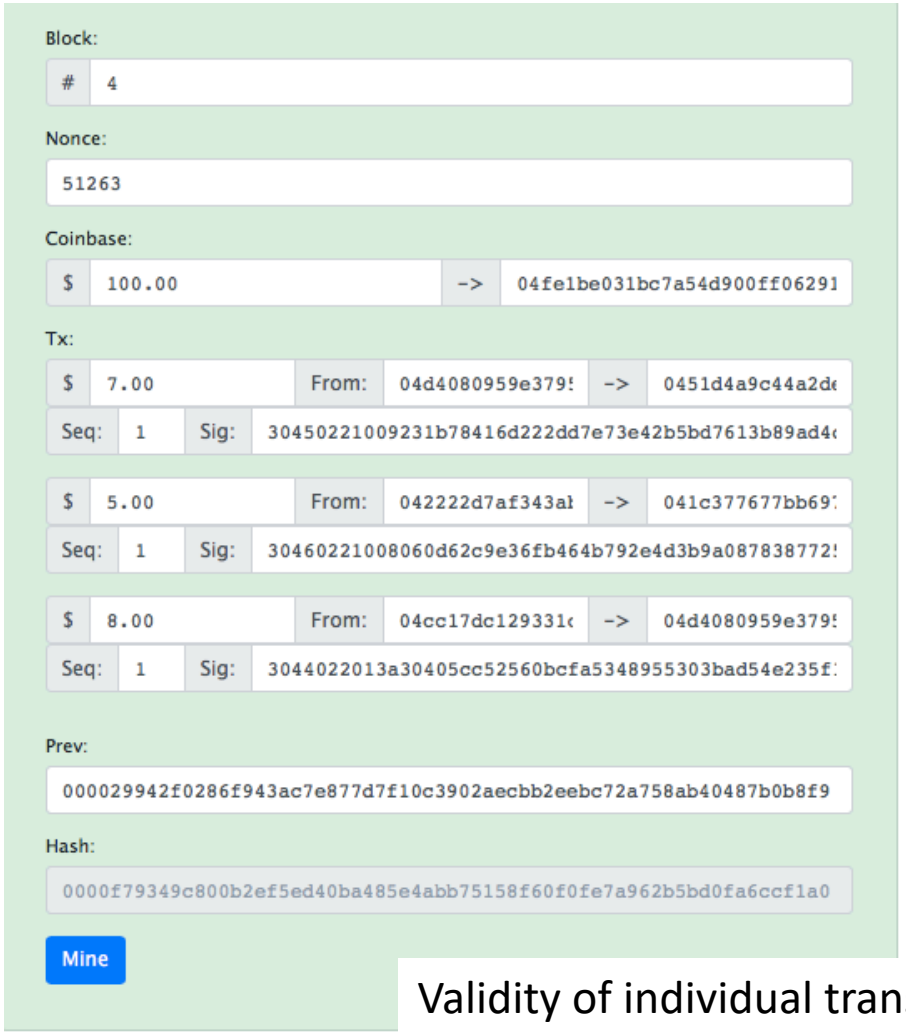


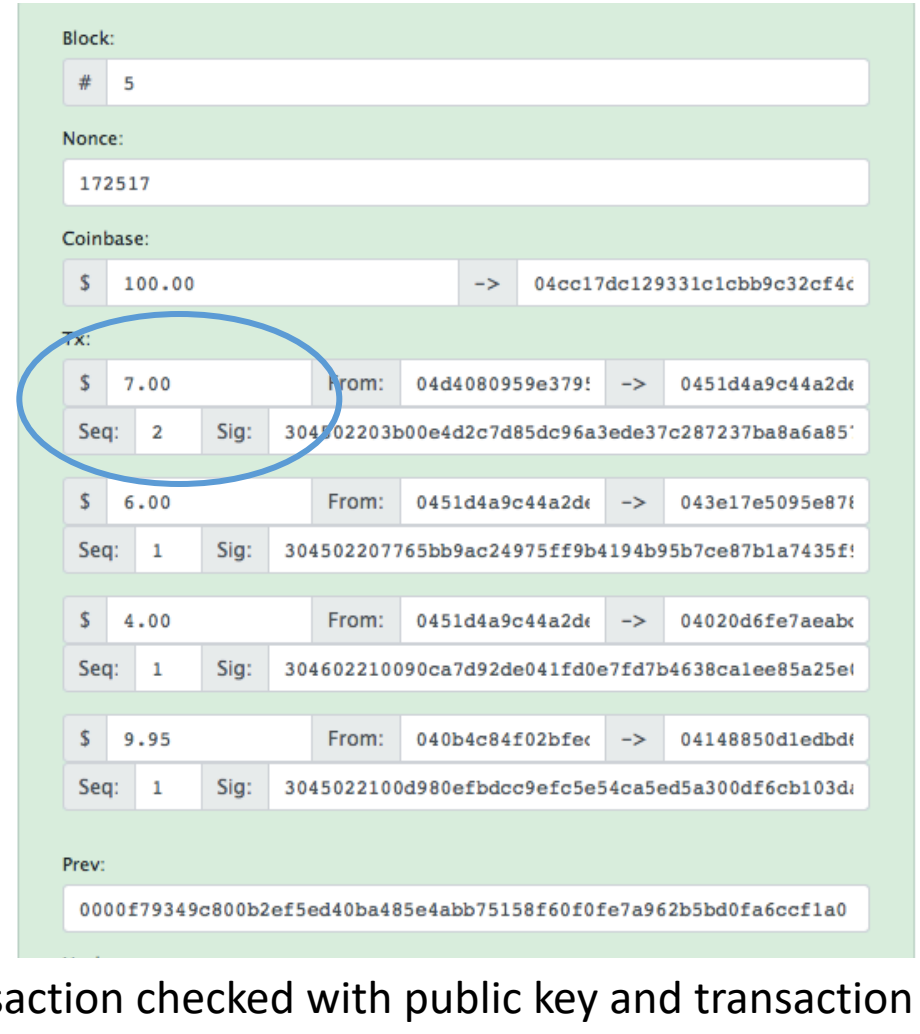
Identifying invalid (modified) transaction



Validity of block transaction checked with hash and data.

Cryptographically Signed Transaction

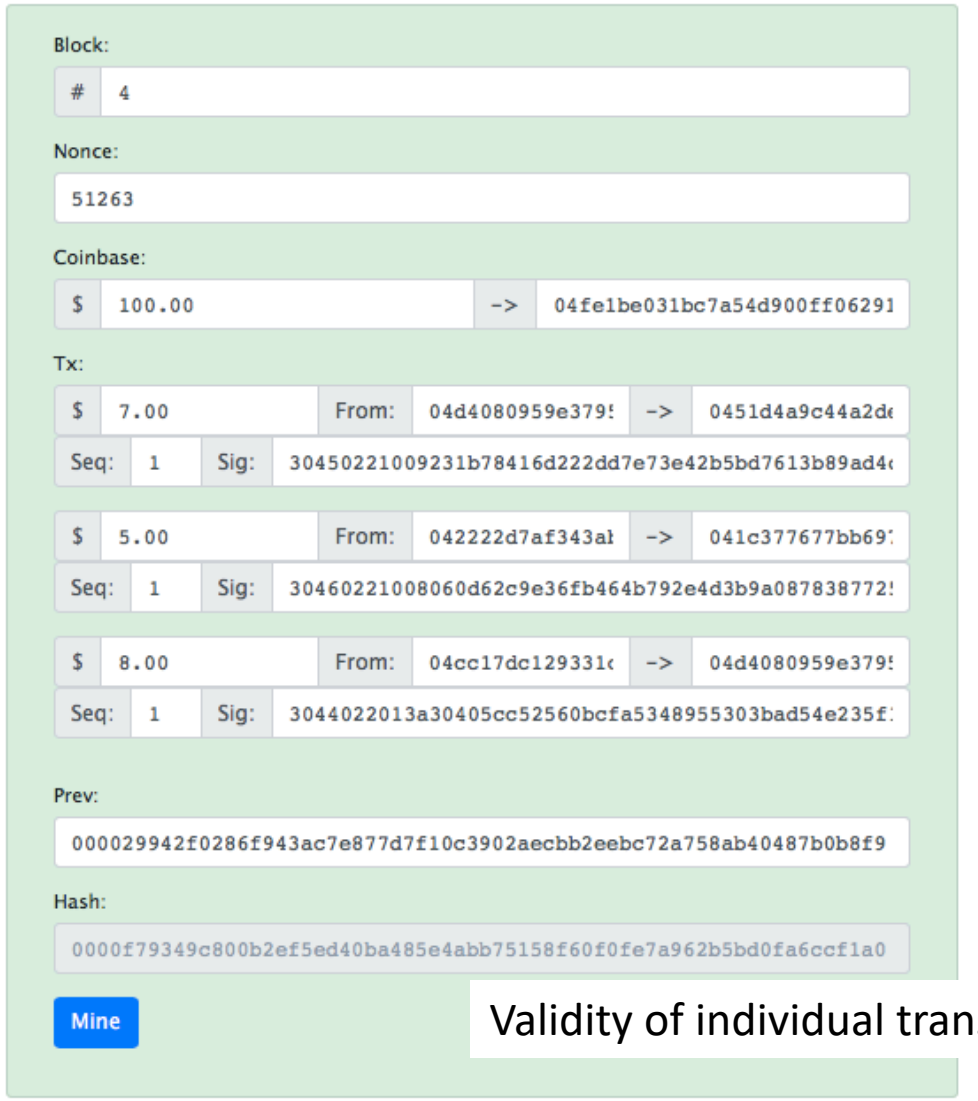




Validity of individual transaction checked with public key and transaction data.

Validity of block transaction checked with hash and data.

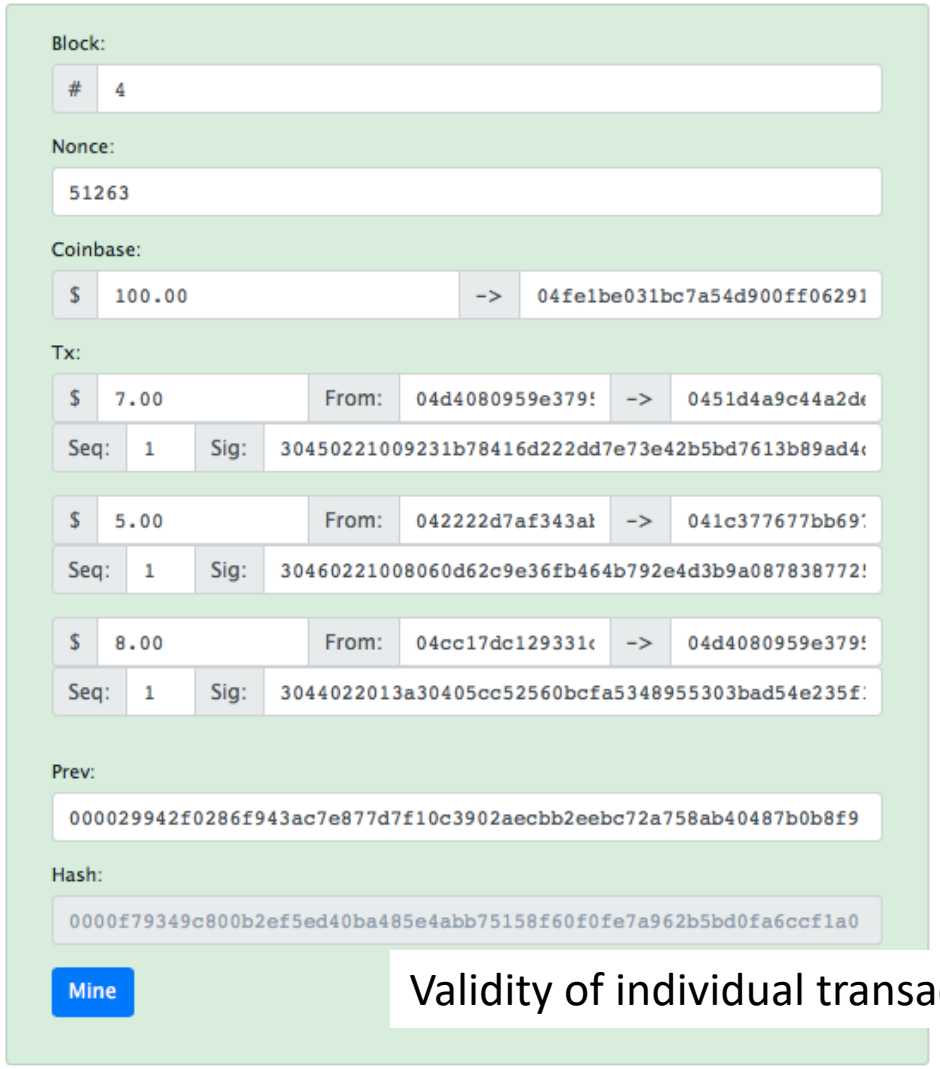
Modified Transaction on Blockchain

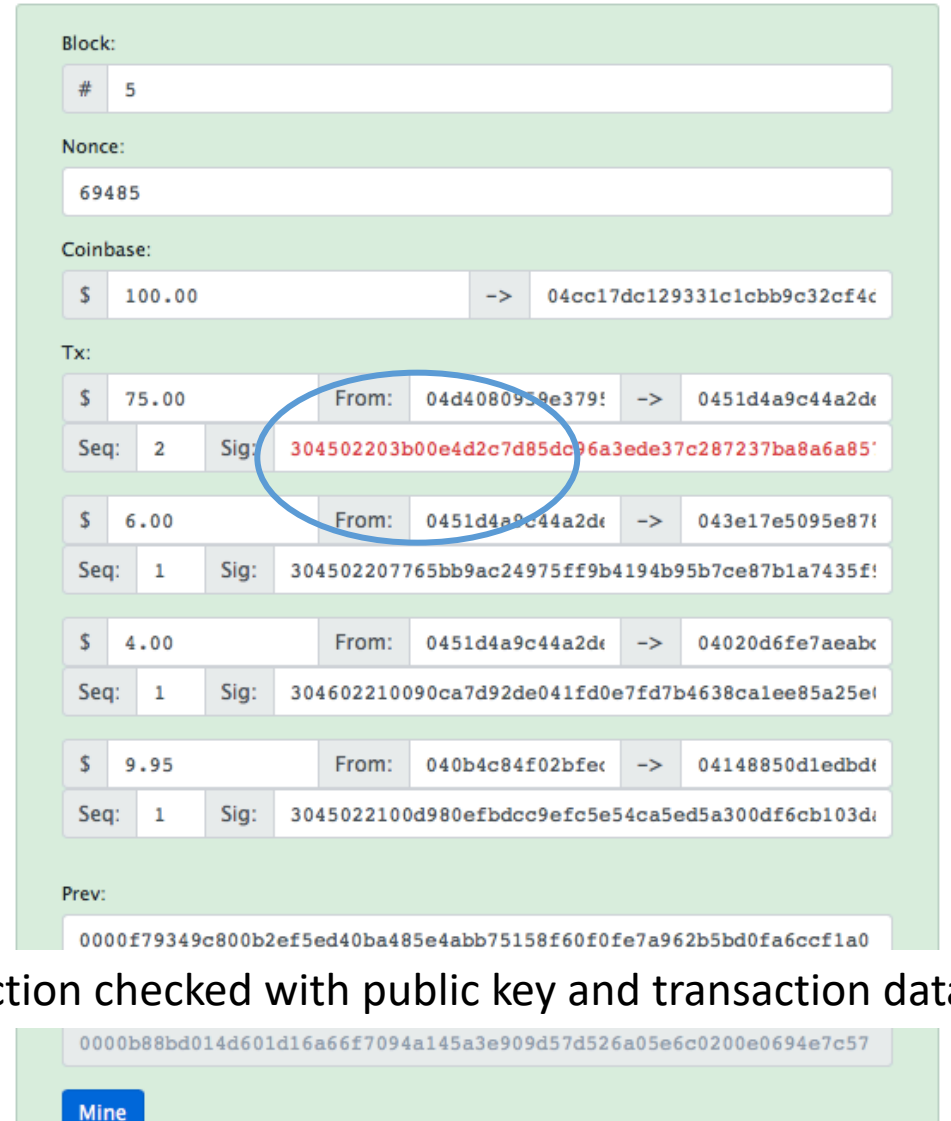


Validity of individual transaction checked with public key and transaction data.

Validity of block transaction checked with hash and data.

Remined/modified Transaction





Validity of individual transaction checked with public key and transaction data.