CS 419: Computer Security

Week 9: Blockchains & Bitcoin

Paul Krzyzanowski

Hash Pointers

Hash Pointers

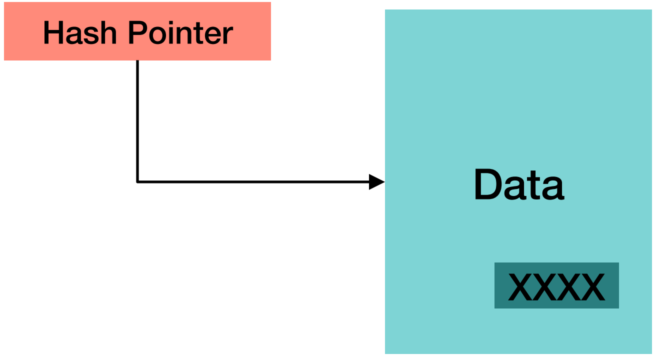
Alternative to pointers in data structures

Hash pointer = { pointer, hash(data) }

pointer = reference that identifies where the object is:
memory location, file name, object ID, server/object, …

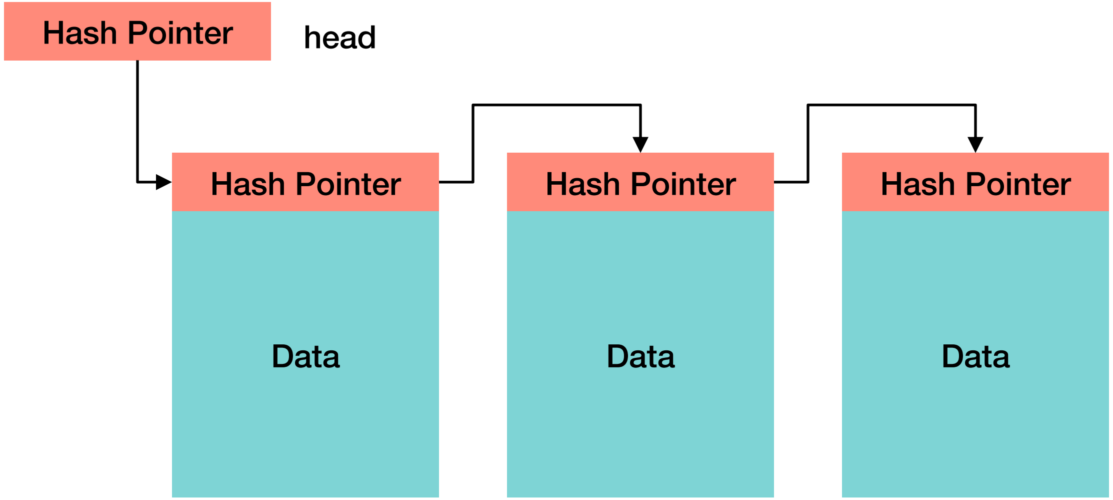
hash(data) = hash function applied to the data being pointed to

Tamper Detection With A Hash Pointer



- **If an attacker modifies data, hash(data) ≠ hash in pointer**
- **This allows us to verify that the information we're pointing to has not changed**
  - Before using that data, do a hash(data) and see if it matches the hash in the hash pointer

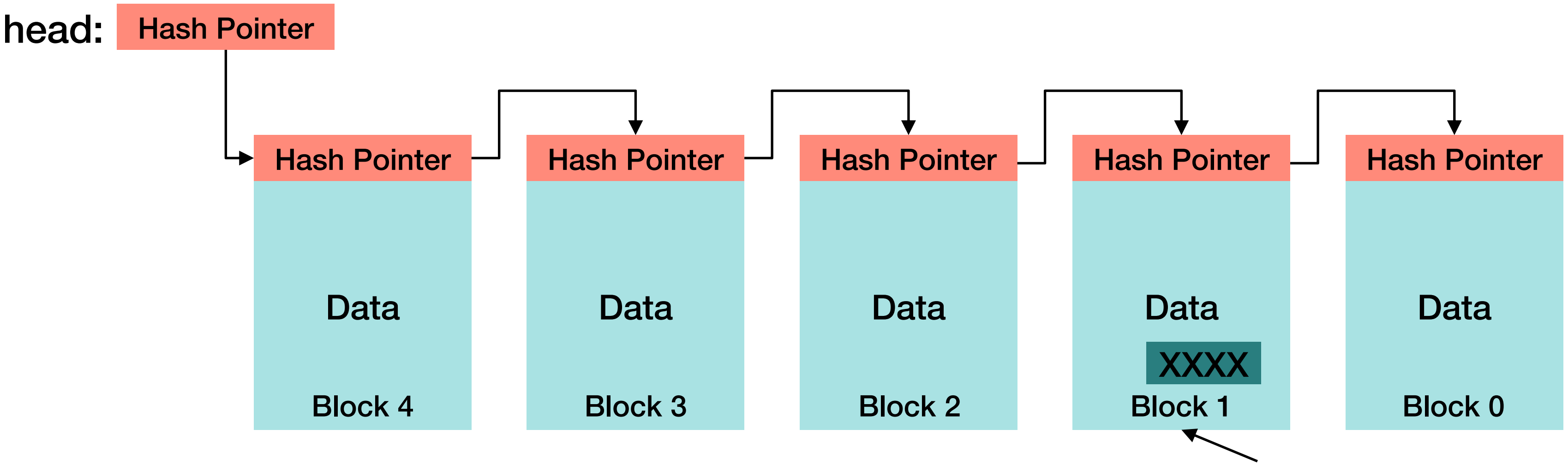
Hash Pointers: Linked Lists = **Blockchain**



- **Add new data blocks to the end of the list**
  - Each hash pointer contains a pointer & a hash of the entire data structure to which it is pointing: the application data and the hash pointer in that structure

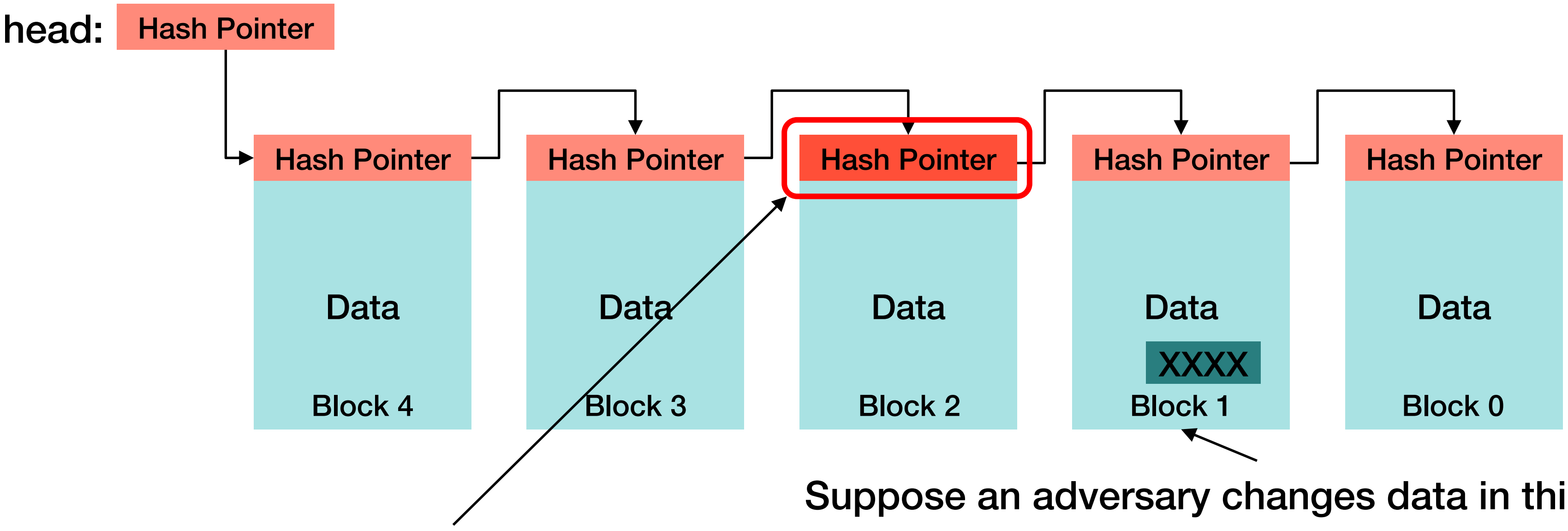
Tamper Evident Log = Blockchain

Tamper detection



Suppose an adversary changes data in this block

Tamper detection

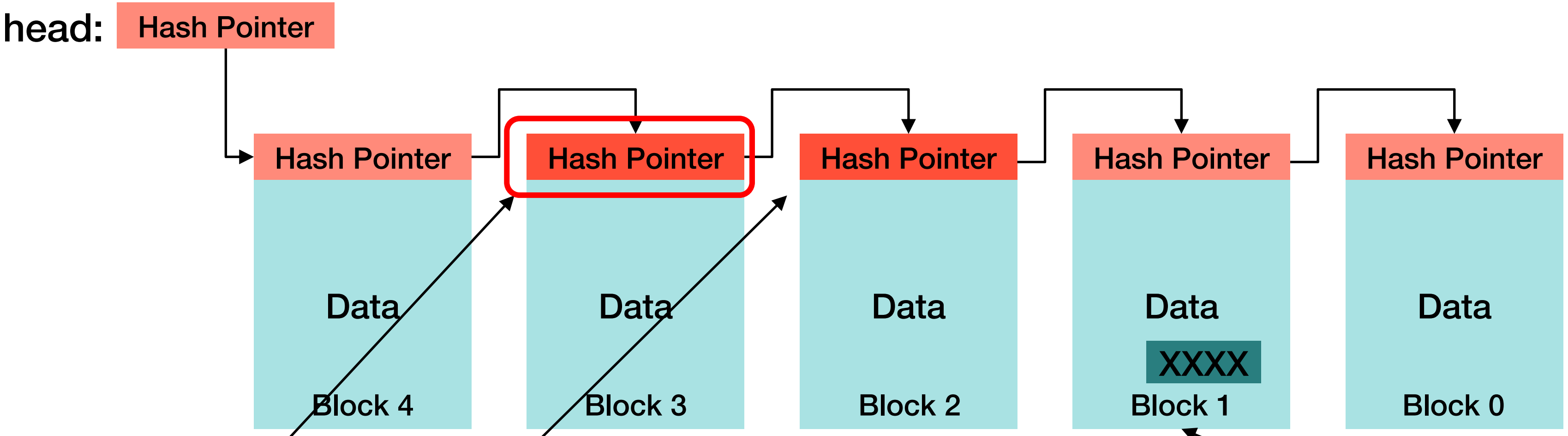


Suppose an adversary changes data in this block

Then this hash pointer needs to be changed

The adversary needs to update the hash in the pointer to match the hash of the modified block

Tamper detection



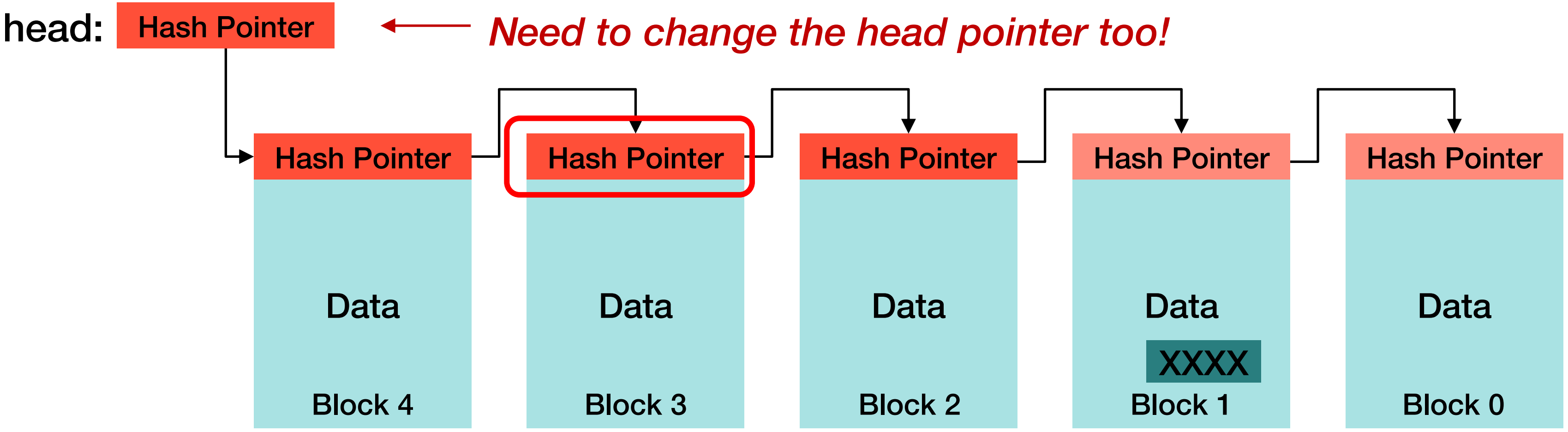
Suppose an adversary changes data in this block

Then this hash pointer needs to be changed

The adversary needs to update the hash in the pointer to match the hash of the modified block

The hash in this pointer is now invalid, so it needs to be updated

Tamper detection

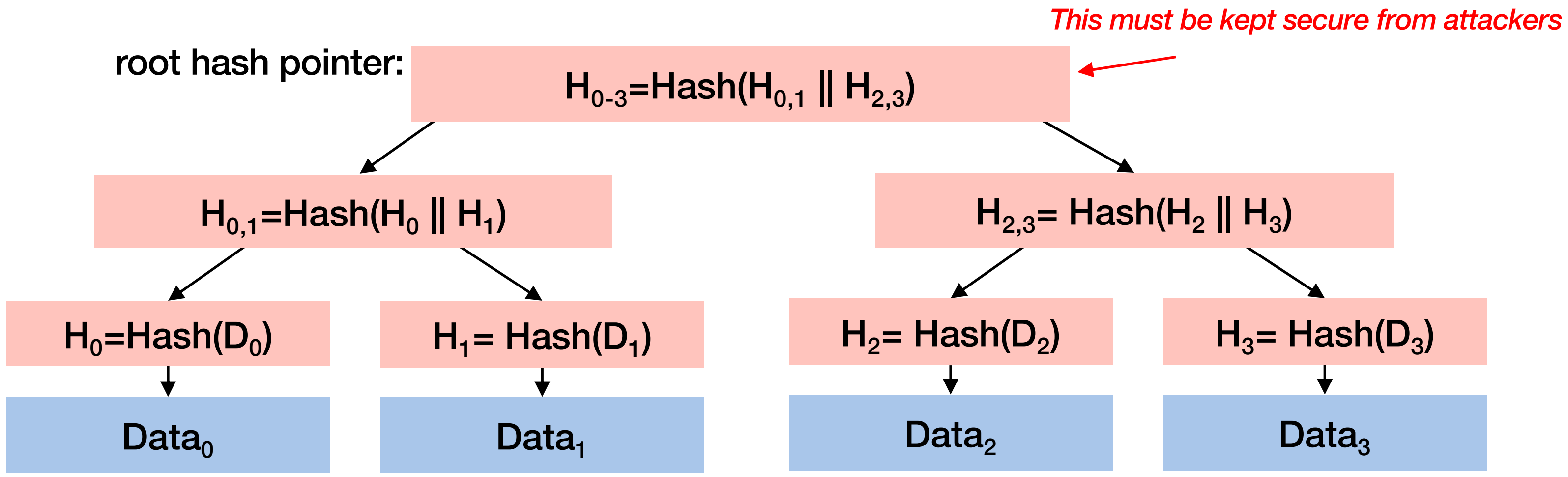


- **The adversary will have to change all hash pointers back to the head**
- **If we can keep the head of the list safe so an adversary cannot modify it, then we can always detect tampering**

It takes less effort to modify newer blocks than older ones

Merkle Trees: Binary trees with hash pointers

Merkle Tree hash pointer = { left_subtree, right_subtree, *hash*(left || right) }


This must be kept secure from attackers
root hash pointer:
H_{0-3}=Hash($H_{0,1}$ || $H_{2,3}$)
$H_{0,1}$=Hash(H_0 || H_1)
$H_{2,3}$= Hash(H_2 || H_3)
H_0=Hash(D_0)
H_1= Hash(D_1)
H_2= Hash(D_2)
H_3= Hash(D_3)
$Data_0$
$Data_1$
$Data_2$
$Data_3$

- Tamper-resistant tree structure
- Only need to examine O($\log_2 n$) hashes to validate a data block belongs to the tree

a || *b* means *a* concatenated with *b*

Merkle Trees (Hash Trees): Uses

- **Commonly used in peer-to-peer data updates**
- **You receive updated content from an untrusted peer**
  - Validate that the data blocks have not been damaged or modified
  - Don't need to wait for all content to be downloaded
  - Root hash should be obtained from a trusted place (or signed)
- **Used in**
  - Version control systems: Git, Mercurial
  - File systems (to detect data damage): ZFS, IPFS
  - Distributed databases: Cassandra, Dynamo, Riak
  - Backup systems: Tahoe-LAFS
  - Decentralized websites: ZeroNet
  - Cryptocurrency: Bitcoin & Ethereum (maybe others)

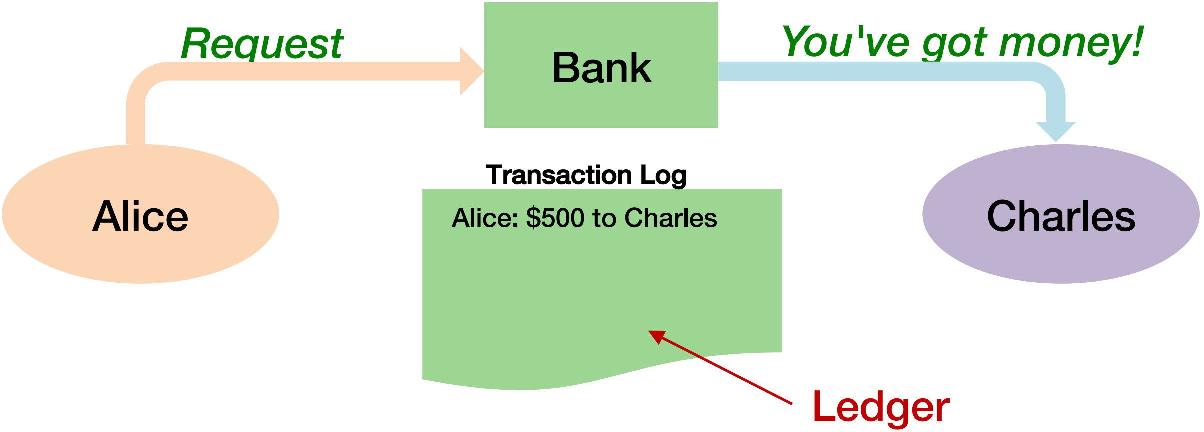
Bitcoin

Bitcoin Cryptocurrency

- **Introduced in 2009 – anonymously by Satoshi Nakamoto**
- **First blockchain**
- **Designed to be public**
  - Anyone can participate in the system & use it
  - Users are anonymous
- **Currency that is totally separate from any sovereign government**
  - Anyone can create money!

Traditional Payments

- **Suppose Alice wants to pay Charles**
  - Send a message to the bank: *transfer $500 from Alice to Charles*
- **Bank is a trusted third party**
  - Owns register of activity & account balances
  - Only the bank can manipulate the data
  - Also – banks control supply of money



Centralized systems

Transactions are simply modifications to the bank's database

- **We can simply**
  - Subtract $500 from Alice's account
  - Add $500 to Charles' account
- **Having a log is just nice for auditing but not necessary**

Problems?

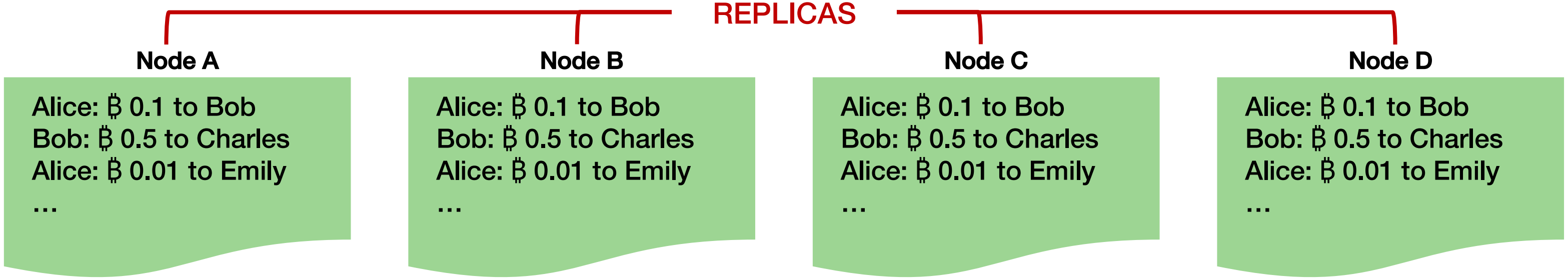
This is a centralized system

We trust the bank – it is a trusted third party

- What if the bank disappears?
- What if the banker makes a mistake?
- What if the banker is corrupt?

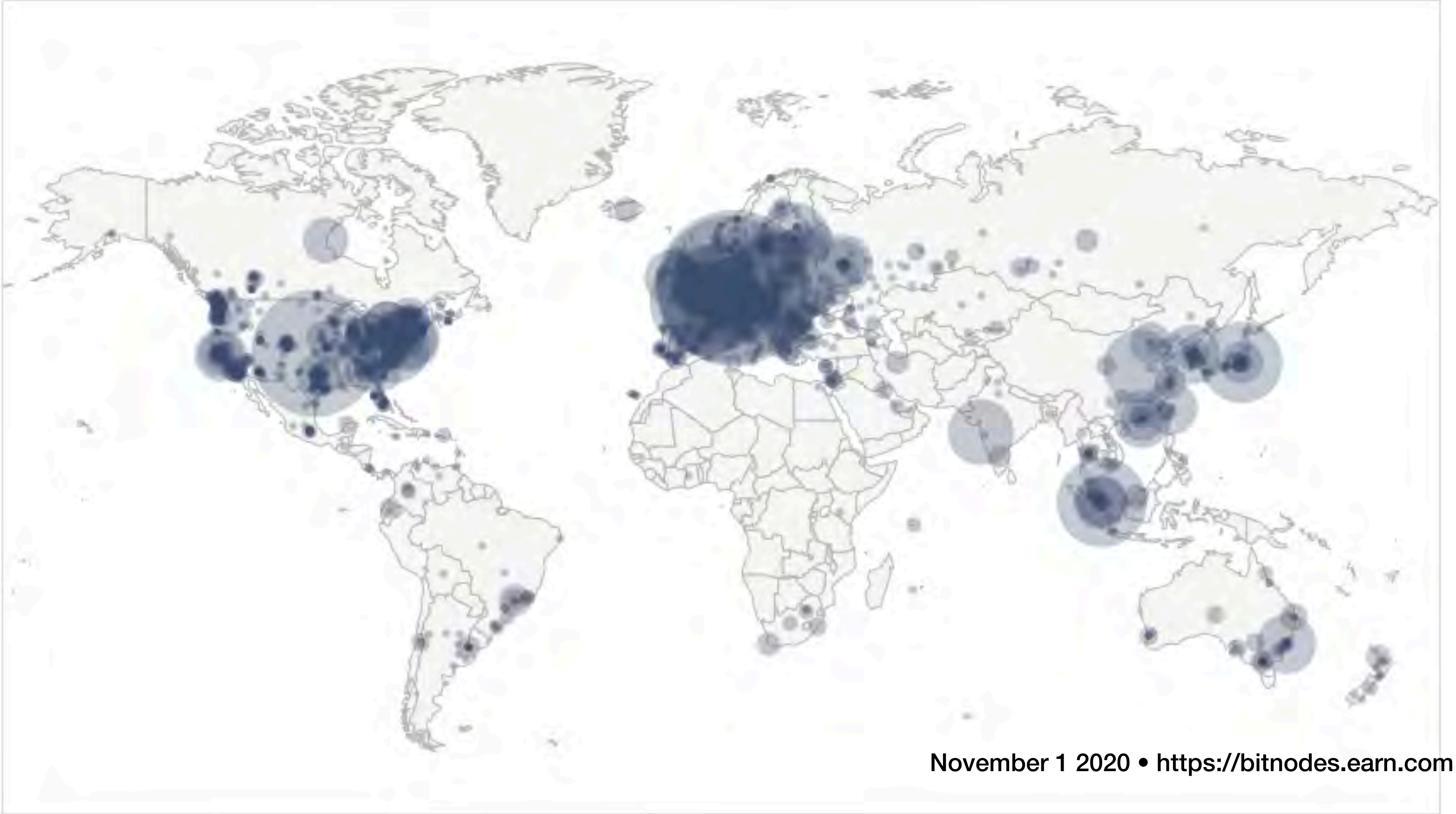
Decentralized Solution – Bitcoin

- **Blockchain = ledger = complete list of ALL transactions**
  - Since Bitcoin was started in January 2009
  - 307,833 MB as of Nov 1, 2020 (See https://www.blockchain.com/en/charts/blocks-size for the current size of the ledgers)
- **Complete copies of the ledger are replicated around the world**
  - 11,177 nodes (Nov 1, 2020) – all peers – running identical software (See https://bitnodes.earn.com)
  - There is **no master node** or master version
- **New systems can do a DNS query for well known peers**
  - Names hardcoded in source (DNS seeds)
  - Return list of IP addresses of bitcoin nodes
  - Then use peer discovery process to find others



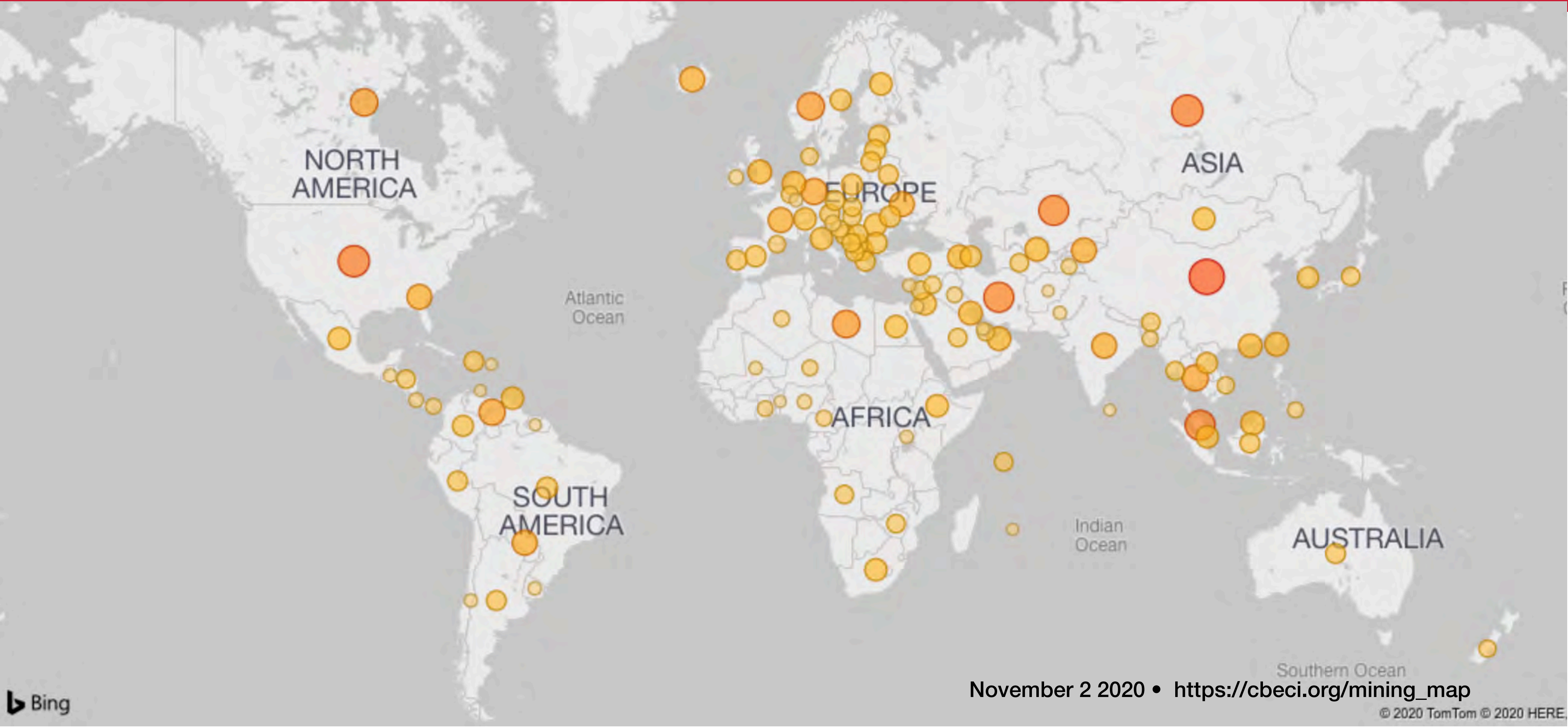
Alice: ฿ 0.1 to Bob
Bob: ฿ 0.5 to Charles
Alice: ฿ 0.01 to Emily
…

Global Bitcoin Nodes



November 1 2020 • https://bitnodes.earn.com

Global Bitcoin Nodes – compute power



November 2 2020 • https://cbeci.org/mining_map

Identities

- **User creates a {public, private} key pair that defines her wallet**
  - 256-bit Elliptic Curve Digital Signature Algorithm (ECDSA) used
  - **The wallet is just local place for users to store these keys**
    - Wallets may store a transaction list but that's just for user records – the bitcoin network doesn't care
- **Bitcoins are associated with keys, not users**
  - Users are anonymous
  - A user's ID is the public key – anonymous – no association to name
  - The user's identity is called their **address**
  - Users may have multiple keys & multiple addresses
- **Every transaction is signed with the creator's private key**
  - Transaction identifies the user by the public key and can be verified
  - We know only the person with the corresponding private key could have created the request

Nobody to call if you lose your private key!

Bitcoin address = hash(public_key)

Bitcoin uses ECDSA: Elliptic Curve Digital Signature Algorithm

A user creates one or more identities = { private, public } key pairs

You can create an identity (address) for each new transaction

How Bitcoin addresses are created*

1. Generate an ECDSA public, private key pair
2. Create a SHA-256 hash of the public key
3. Perform a RIPEMD-160 hash on that
4. Add a version byte in front of the result
5. Perform a SHA-256 hash on the result … and a SHA-256 hash on that
6. First 4 bytes of the result = address checksum
7. Add 4 bytes from [6] to the end of the RIPEMD-160 hash from [4]
8. Convert the byte to a base-58 string using Base58Check encoding.
   This produces a 20-byte *address*

***You don’t have to know this.**

Addresses vs. keys

- **Spending: Bob wants to send Alice 5 bitcoins**
  - Bob creates a transaction with a **digital signature** using his private key
  - Presents his public key along with the transaction
  - Any receiving node can validate that the transaction was signed by someone with the corresponding private key
  - The destination of the money is Alice's **address**
- **Addresses are not accounts**
  - They only receive funds
  - You can use those funds if you prove you know the private key that corresponds to the address
  - If Alice wants to use the coins she received
    - She creates a new transaction with her public key & a digital signature
    - Any node can validate that the address belongs to her
    - No node can figure out Alice's public key just by looking at the address

https://learnmeabitcoin.com/guide/public-key-hash160

Transactions: Inputs

If Alice wants to send some bitcoin to Charles

- She creates a **transaction** and sends it to one or more bitcoin nodes
- A node tells its peers about the transaction
- Within ~5 sec. every peer on the network has it
- The transaction is currently **unconfirmed**

A blockchain is NOT a database – it's a list of transactions

- There are no accounts to query
- Alice needs to provide links to previous transactions that will add up to at least the required amount – these are **inputs**

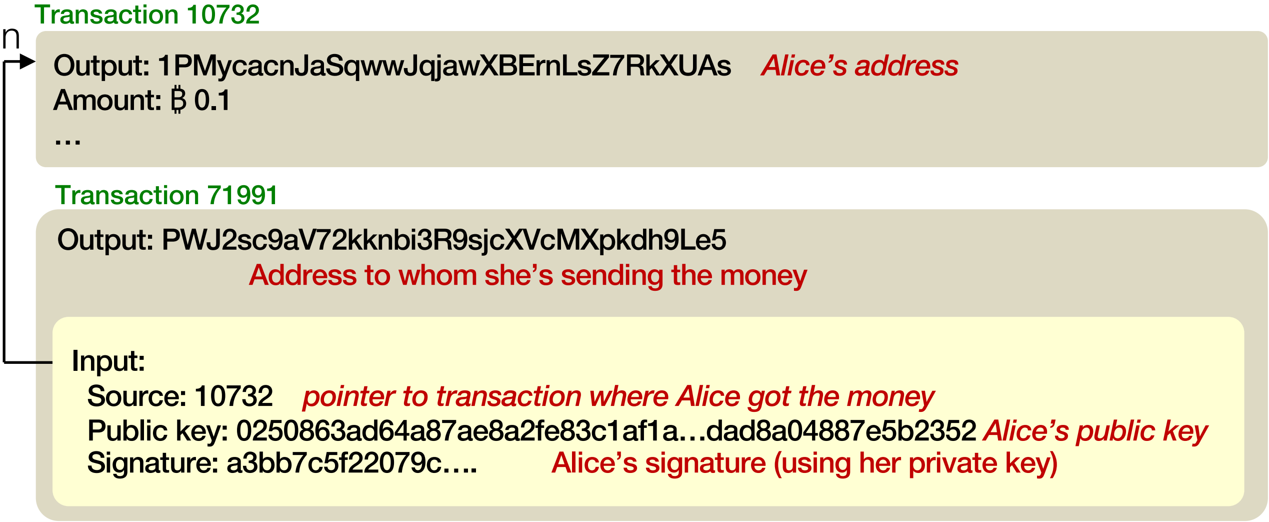
A node verifies inputs

- Make sure they have not been used by another transaction—this would be **double spending**
- Make sure there is sufficient money in the inputs

Addresses vs. keys – Inputs and Outputs

If Alice wants to use the coins she received:

- She creates a new transaction with her public key & a digital signature
- Any node can validate the signature using her public key



Transaction 10732

Output: 1PMycacnJaSqwwJqjawXBErnLsZ7RkXUAs *Alice's address*
Amount: ฿ 0.1
…

Transaction 71991

Output: PWJ2sc9aV72kknbi3R9sjcXVcMXpkdh9Le5
Address to whom she's sending the money

Input:
Source: 10732 *pointer to transaction where Alice got the money*
Public key: 0250863ad64a87ae8a2fe83c1af1a…dad8a04887e5b2352 *Alice's public key*
Signature: a3bb7c5f22079c…. Alice's signature (using her private key)

The transaction can be validated by validating each input:

1. Validate the signature using Alice's public key (in the transaction). This proves that whoever created the signature has the private key corresponding to the public key.
2. Hash the public key in the transaction to create the address – see if it matches the address in the referenced transaction
3. No node can figure out Alice's public key just by looking at the address

https://learnmeabitcoin.com/guide/public-key-hash160

Transactions: Inputs & Outputs

A transaction contains:

1. **One or more inputs: transaction IDs & address where coin comes from**
   - *Contains signature & public key*
   - An input is a reference to the output from a previous transaction
2. **Output: whom the money goes to – destination address & amount**
3. **Change: owner's address & amount**
   - Every input must be completely spent
   - Any excess **change** can be generated as another output to the owner of the transaction
4. **Transaction fee (anywhere from 10¢ to a few $ per transaction – currently ~ $13)**
   - There's a limited amount of space (1 MB) in a block. A transaction is about 250 bytes. To get your transaction processed quickly, you need to outbid others.

The amount of bitcoin you own is the set of transactions in the system that are outputs to your address but have NOT been used as inputs in any transaction

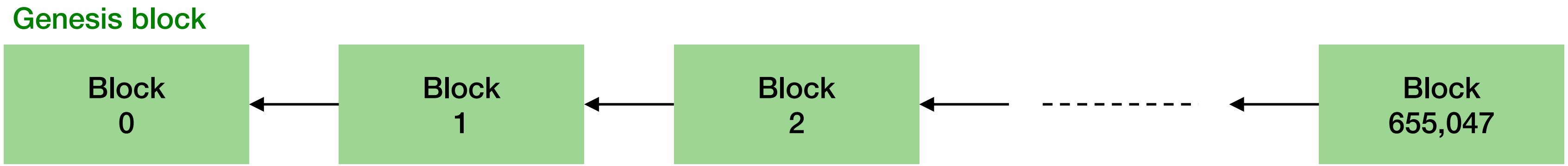
Blocks

Transactions are grouped into blocks

– Each block holds ~2,220 transactions @250 bytes and is ~1.25 MB in size

Bitcoin ledger = linked list of chronologically-ordered blocks

Approximately every 10 minutes, a new block of transactions is added to the blockchain



Each block has

– A link to the previous block

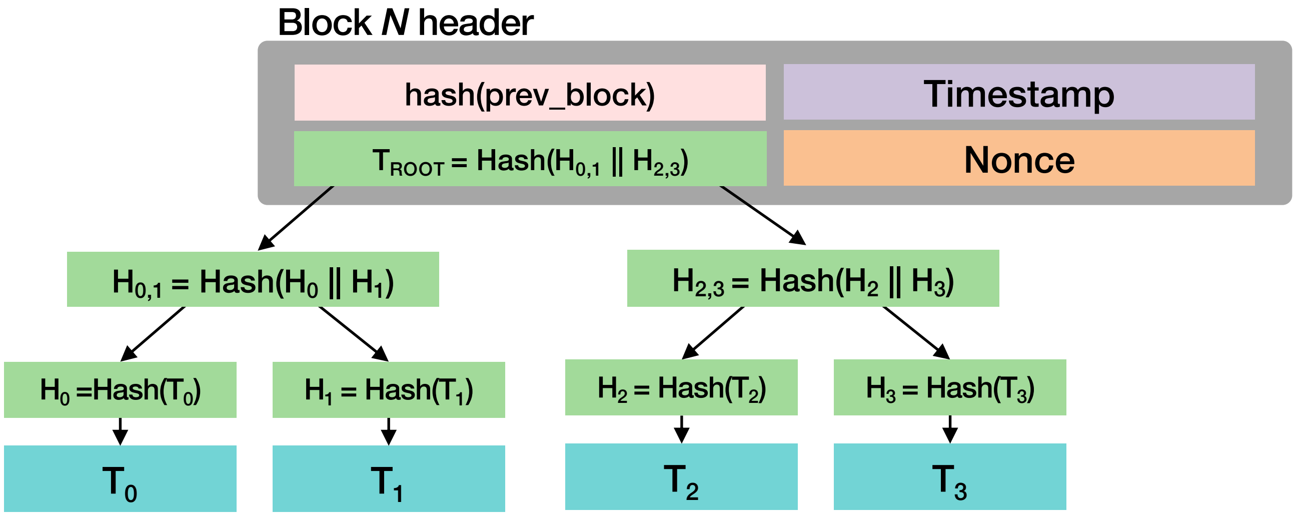
– SHA-256 hash of the previous block

This creates the blockchain: hash pointers – a tamper-evident log

Transactions in blocks: Merkle trees

Transactions *<u>within</u>* a block are stored in a Merkle tree

- Binary tree of hash pointers
- Using a tree makes it easy to find one of thousands of transactions
- A Merkle tree makes it easy to check if the transaction is valid



Agreement & adding blocks

Each node groups transactions into a block & can propose it as the next block in the blockchain

- Transactions can reach nodes in a different order
- We want all nodes **to agree on the sequence** of blocks in the blockchain

A linked list of hash pointers (blockchain) is a tamper-proof structure

- If the contents of any block are modified, then the hash pointer that points to the block will not be valid (the hash in the pointer won't match)
- ***But can't anyone change the hash pointers?***

  We might want to use signed pointers but there's no central authority (no trusted party), so that won't work

Let's create a system where
(a) Everyone can agree on the sequence of blocks
(b) That sequence cannot be modified

To add a block to the chain, the hash of the block must meet a certain requirement

Make block addition challenging: *create a puzzle*

Suppose we want a hash value to have a specific property:

- Example: the hash should start with with "0000"

There is no algorithmic way to do this

Must try lots of variations of the input

But once found –
it is easy for anyone to verify that the data hashes to the result

Just hash the data and see if the hash starts with "0000"

Mining

Solving this "puzzle" is called mining

- A block has a 32-bit field in the block where we can try different numbers
- Try to get the block to hash to a desired output
- The resulting number is called the Proof of Work – *difficult to generate but easy to verify*

We demonstrate that work has been put into figuring out what the value should be to create the desired hash

Everyone in the network can participate in this

- The first system that finds it announces the block to everyone else in the network
- Upon receiving an announcement of a new block:
  1. Each system validates the Proof of Work number against the block
  2. A majority of systems must grant approval
  3. If they do, the block (with the Proof of Work) is made part of the blockchain

What's the puzzle?

- **Bitcoin uses a version of hashcash** (created in 1997)
  - Search for a SHA-256 hash:

    hash(block_header) < target value

  - Choice of ***target_value*** sets the difficulty of the problem

- **Hashed block header contains:**
  - *Version number*
  - *Hash pointer to previous block*
  - *Merkle root hash (hash of all transactions in the block)*
  - *Timestamp when mining started on the block*
  - *Difficulty target (compact format – 4 bytes)*
  - ***N* – nonce: the 32-bit number we modify to get the hash with the properties we need**

Adjusting the target

- **Bitcoin self-tunes so a new block is mined every 10 minutes on average**
- **Proof of Work Target number is adjusted every 2,016 blocks**
  - 2016 blocks × 10 minutes ≈ approximately every two weeks
    - Look at timestamps in the last 2016 blocks
  - Adjust the value to bring it to two weeks
    - If the previous 2016 blocks took > 2 weeks to mine, the number is adjusted to make mining easier (bigger value)
    - If the blocks took less time to mine, the number is adjusted to make mining more difficult
- **Target value**
  - *Roughly* – the number of leading zero bits in the hash output
  - Target is a # with a very large # of leading 0s – currently ~17

See: https://chainbulletin.com/proof-of-work-explained-in-simple-terms/
https://www.investopedia.com/terms/b/bitcoin-mining.asp

Isn’t a 32-bit nonce too small?

A 32-bit nonce field might not be sufficient

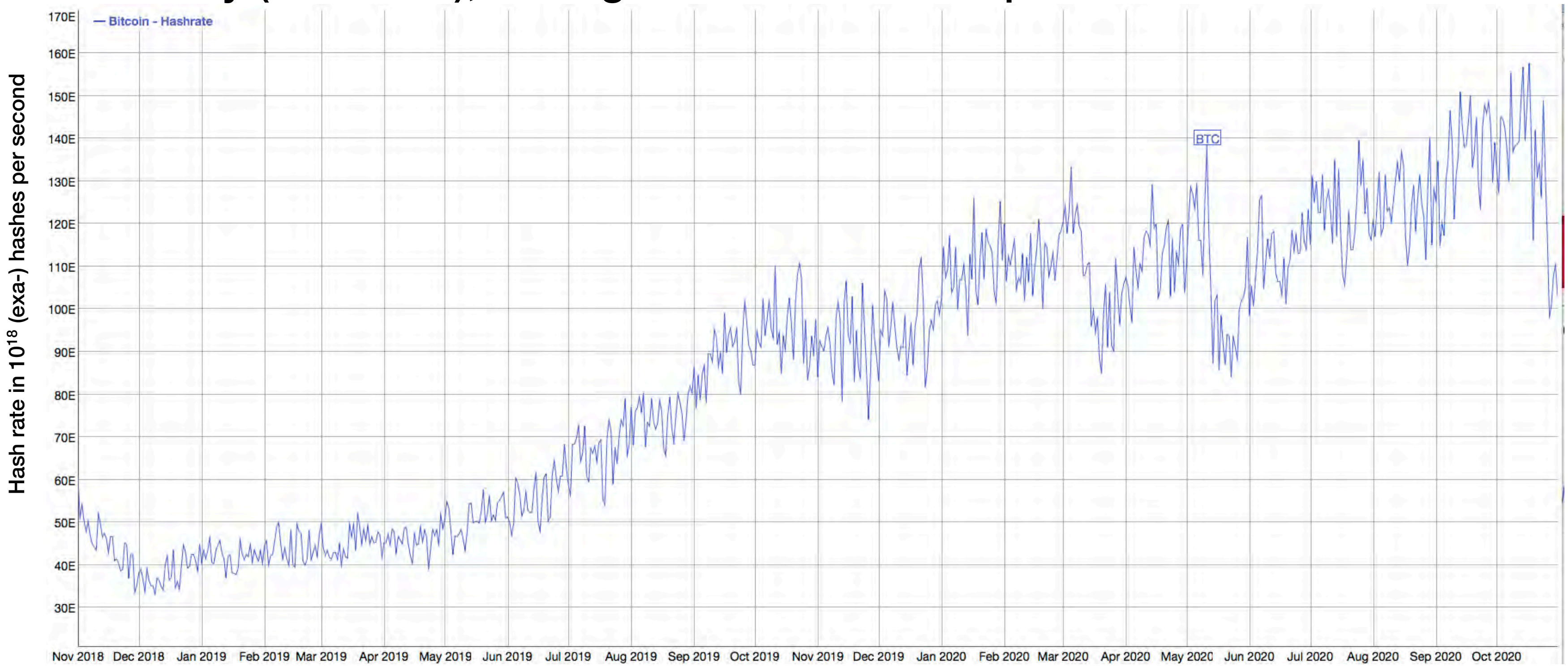
- There might be NO value of the nonce that will produce the right hash

The node then needs to modify other data in the block header & try again

- Make small changes to the timestamp
- New transactions may be added
- If nonce overflows, increment `extraNonce` & reset nonce
  - `extraNonce` is 2-100 bytes
  - Changing `extraNonce` alters the Merkle root hash
    - That needs to be recomputed

How much work is going on?

Currently (Nov 2020), average 113×10^{18} hashes per second



See https://bitinfocharts.com/comparison/bitcoin-hashrate.html#2y

Bitcoin mining

Computing the hash = mining

If you come up with the right answer, you win!

1. You send your block, with the nonce in it, to the whole network
2. Others validate it
3. When a computer validates your block, it adds it to its ledger
4. You get a reward for solving the puzzle! Currently 6.25 BTC
5. You also get paid transaction fees in the transactions you put into the block
6. The block (not transactions!) is confirmed and you get paid
7. Individual transactions may require the confirmation of multiple subsequent blocks
   ... More on this later….

Bitcoin mining

The more hashes you can try, the better your chances of winning

- **You’re competing with every other miner**
- **People moved from CPU-based mining to GPU-based**
  - GPU power approximately = 30 CPUs
  - Then FPGA mining: approximately 3-100x faster than GPUs
  - ASIC mining (application-specific integrated circuit):
    - Special hardware built for hash computation: faster & more power efficient
- **Mining pools = group miners together & share rewards**
  - There are over a dozen large pools for Bitcoin

Mining Hardware

CPU → GPU → FPGA → ASIC

Example:

Bitmain

Antminer S19 Pro

Computes SHA-256 hashes at 110 TH/s

~$3,200

Consumes 3250 watts

Estimated profit: $834.14/year

https://www.asicminervalue.com/miners/bitmain/antminer-s19-pro-110th engadget

It takes an estimated seven nuclear plants to power our bitcoin mining

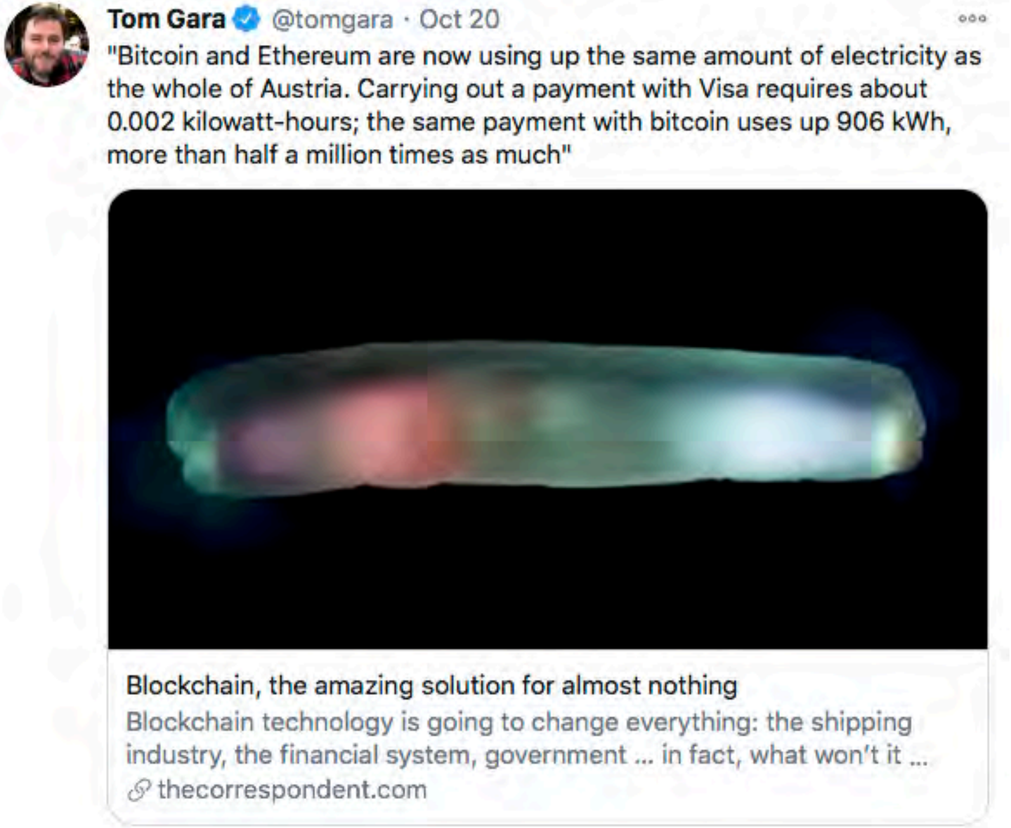
That's 21.8 million solar panels worth of production.

Andrew Tarantola – August 26, 2020

Turns out that plugging a bunch of computers into our electrical grid that do nothing but draw current and hash through algorithms has had some negative environmental impacts. Recent studies suggest that Bitcoin-related power consumption has reached record highs this year — with more than seven gigawatts of power being pulled in the pursuit of the suspect digital currency.

A study from the Cambridge Center for Alternative Finance released on Monday estimates that the global bitcoin mining industry uses 7.46 GW, equivalent to around 63.32 terawatt-hours of energy consumption. The study also notes that miners are paying around $0.03 to $0.05 per kWh this year. Given that a March estimate put the cost to mine a full bitcoin is around $7,500, the average miner still stands to make over $4,000 in profit from the operation.

https://www.engadget.com/it-takes-an-estimated-seven-nuclear-plants-to-power-our-bitcoin-mining-212441059.html



Tom Gara @tomgara · Oct 20

"Bitcoin and Ethereum are now using up the same amount of electricity as the whole of Austria. Carrying out a payment with Visa requires about 0.002 kilowatt-hours; the same payment with bitcoin uses up 906 kWh, more than half a million times as much"

Blockchain, the amazing solution for almost nothing

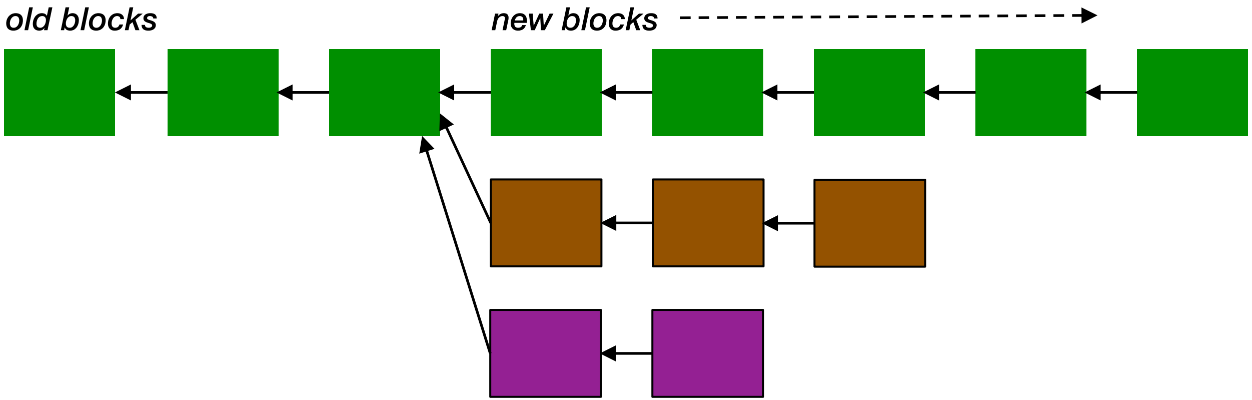
Blockchain technology is going to change everything: the shipping industry, the financial system, government … in fact, what won't it …

thecorrespondent.com

Competing chains

What if a malicious participant wants to modify an old transaction?

- It will need to modify an old block
- And recompute the Proof of Work (which takes a lot of effort) for the block and *each successive block* (tons of work)
- The participant will be creating another chain in the blockchain



Competing chains

BUT:

- One malicious participant will not be able to catch up with the cumulative work of all the others
- It is expected that some nodes will occasionally have different versions
- **Length of chain** = **score**

If we two versions of the blockchain,
we select the one that was the hardest to generate (= longest chain)

Blockchain rules state that

The longest chain in the network is the correct one

If a participant receives a higher-scoring version, it overwrites its blockchain with the better data & transmits updates to peers

Producing a longer ledger than the current one requires computing power that competes with the rest of the entire network

51% Attack

If a participant has the majority of the hash rate, the protocol will fail

Blockchain works only because of the assumption that the ***majority*** of participants are honest

To double-spend a bitcoin:

- You would need to rewrite the blockchain (change past transactions)
- An attacker would need to control more than 50% of computing capacity
  - This is a lot: as of 12/17, The Economist estimates "*bitcoin miners now have 13,000 times more combined number- crunching power than the world's 500 biggest supercomputers*"
  - Even if someone tried to do this attack, they'd likely only modify transactions in the past few blocks
- Keeping history of all transactions among all participants allows anyone to check for double spending

Confirming transactions

A transaction is *confirmed* after *N* number of additional blocks are added to the blockchain

– Large values of *N* are recommended for high-value transactions

The more blocks are added after a transaction, the more difficult it is to modify it

Bitcoin Confirmation Recommendations

1:	Small payments <$1,000
3:	Deposits and payments of $1,000-$10,000
6:	Large payments $10k-$1M
60:	Payments >$1M

Higher values of *N* mean that an attacker will need to recompute *N+1* Proof of Work values to modify the blockchain

– Computationally not feasible

https://www.buybitcoinworldwide.com/confirmations/

Incentives

Computing the Proof of Work takes a lot of work – *why do it?*

To get earn bitcoin:

– First participant to compute the Proof of Work gets rewarded with bitcoin
– BUT … only after another 99 blocks have been added to the ledger
– This gives miners an incentive to participate & validate transactions

Reward is decreasing (*assumption: bitcoins will be more valuable*)

– 50 bitcoins for the first 4 years since 2008
– 25 bitcoins from 2012-2015
– 12.5 bitcoins from block #420,000 July 9, 2016 – 2019
– 6.25 bitcoins at block #630,000 – around May 24, 2020

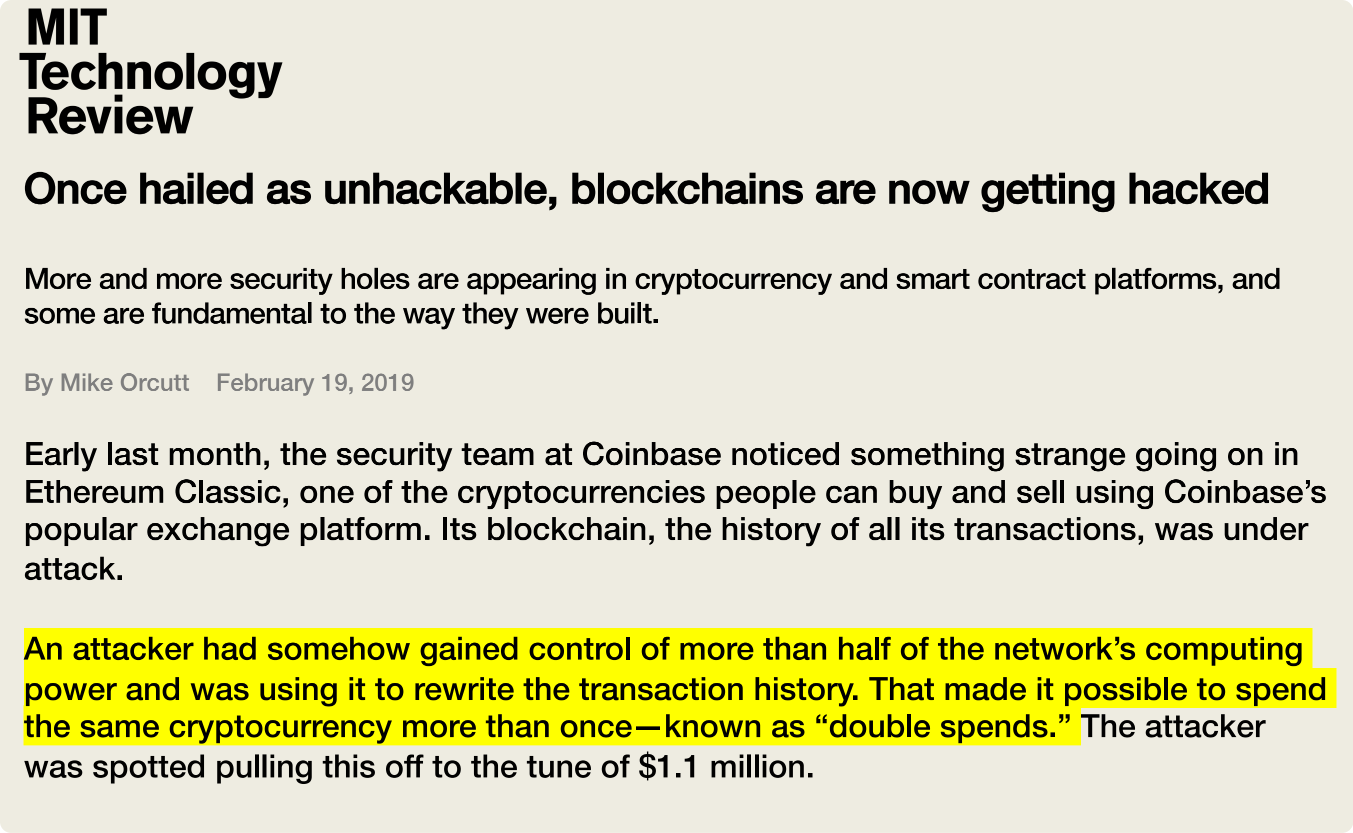
Eventually there will be a maximum of ~21 million bitcoins

There are also transaction fees even if the block reward = 0

Centralization

- **Anyone can run a bitcoin node**
  - Requires a good chunk of disk space but is accessible
  - Highly decentralized
- **Mining**
  - Anyone can mine but requires a lot of computing power
  - Not as decentralized as we'd like
- **Software development/support**
  - Open but there's a core set of trusted developers – not really decentralized
  - Bugs may be fixed … but transactions cannot be undone
- **In theory**
  - Teams of sneaky developers may be able to mount an attack
  - Mining pools may try to mount a 51% attack
  - Both scenarios highly unlikely today

51% attack: difficult, not impossible



MIT
Technology
Review

Once hailed as unhackable, blockchains are now getting hacked

More and more security holes are appearing in cryptocurrency and smart contract platforms, and some are fundamental to the way they were built.

By Mike Orcutt   February 19, 2019

Early last month, the security team at Coinbase noticed something strange going on in Ethereum Classic, one of the cryptocurrencies people can buy and sell using Coinbase's popular exchange platform. Its blockchain, the history of all its transactions, was under attack.

An attacker had somehow gained control of more than half of the network's computing power and was using it to rewrite the transaction history. That made it possible to spend the same cryptocurrency more than once—known as "double spends." The attacker was spotted pulling this off to the tune of $1.1 million.

https://www.technologyreview.com/s/612974/once-hailed-as-unhackable-blockchains-are-now-getting-hacked/

THE COINTELEGRAPH
future of money

Two Miners Purportedly Execute 51% Attack on Bitcoin Cash Blockchain

Max Boddy May 25, 2019

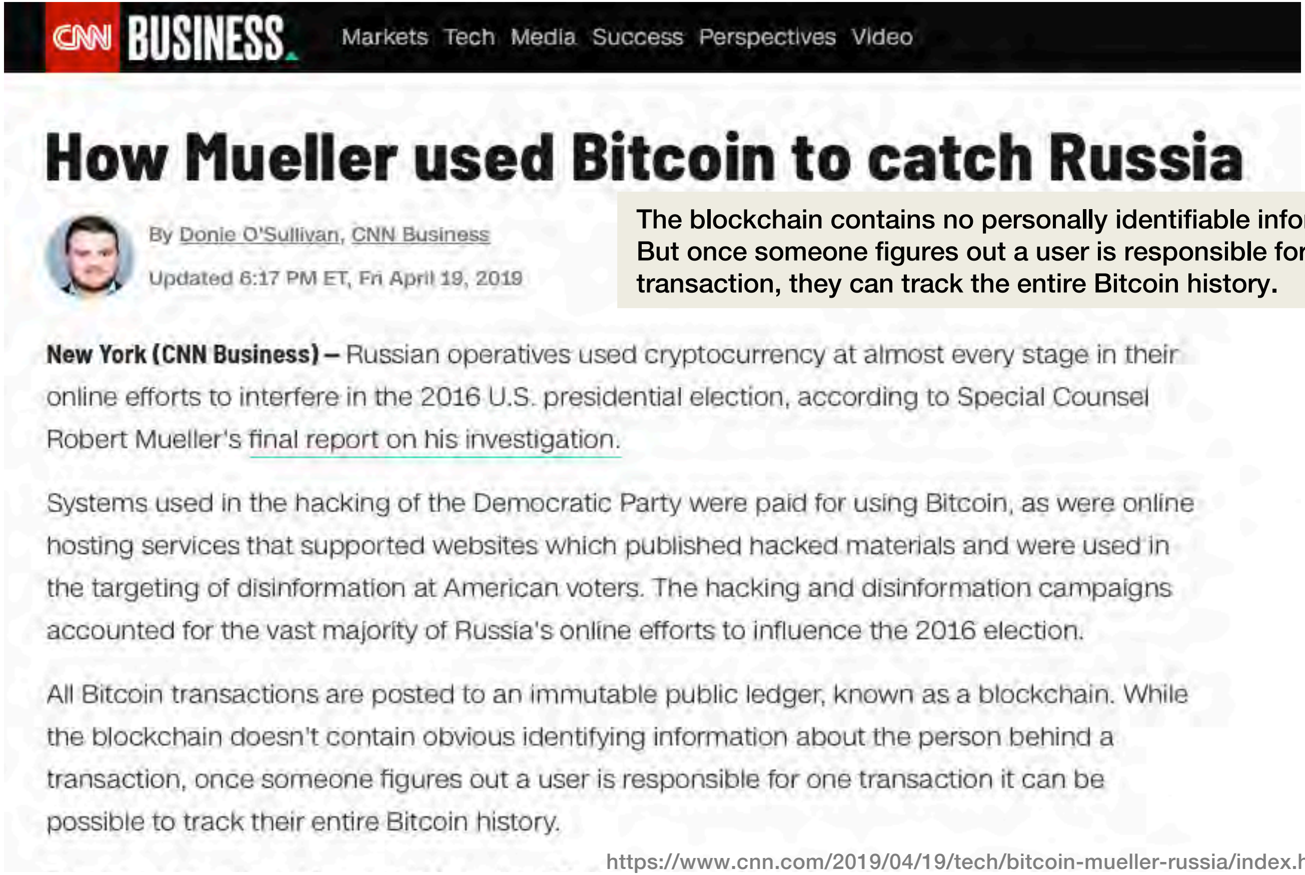
Two miners have reportedly executed a 51% attack on the bitcoin cash (BCH) blockchain, according to tweets by Cryptoconomy Podcast host Guy Swann on May 24.

A 51% attack occurs when someone controls the majority of mining power on a Proof-of-Work blockchain network. This means that the majority block verifier can prevent other users from mining and reverse transactions.

While many have assumed that a 51% attack would be carried out with malicious intent, the above case happened as the two mining pools attempted to prevent an unidentified party from taking some coins that — due to a code update — were essentially “up for grabs.”

https://cointelegraph.com/news/two-miners-purportedly-execute-51-attack-on-bitcoin-cash-blockchain

Achieving anonymity is difficult



How Mueller used Bitcoin to catch Russia

By Donie O'Sullivan, CNN Business

Updated 6:17 PM ET, Fri April 19, 2019

New York (CNN Business) – Russian operatives used cryptocurrency at almost every stage in their online efforts to interfere in the 2016 U.S. presidential election, according to Special Counsel Robert Mueller's final report on his investigation.

Systems used in the hacking of the Democratic Party were paid for using Bitcoin, as were online hosting services that supported websites which published hacked materials and were used in the targeting of disinformation at American voters. The hacking and disinformation campaigns accounted for the vast majority of Russia's online efforts to influence the 2016 election.

All Bitcoin transactions are posted to an immutable public ledger, known as a blockchain. While the blockchain doesn't contain obvious identifying information about the person behind a transaction, once someone figures out a user is responsible for one transaction it can be possible to track their entire Bitcoin history.

The blockchain contains no personally identifiable information. But once someone figures out a user is responsible for one transaction, they can track the entire Bitcoin history.

https://www.cnn.com/2019/04/19/tech/bitcoin-mueller-russia/index.html

A single anonymous market manipulator caused bitcoin to top $20,000 two years ago, study shows

Michael Sheetz
November 4, 2019

One of the SEC's top worries is that crypto is subject to manipulation

A forensic study found that tethers, a digital currency, being traded for bitcoins, revealed a pattern of manipulation during the 2017 cryprocurrency boom.

"Almost the entire price impact can be attributed to this one large player," finance professors John Griffin and Amin Shams wrote.

A forensic study on bitcoin's 2017 boom has found that nearly the entire rise of the digital currency at the time is attributable to "one large player," although the market manipulator remains unidentified.

Finance professors John Griffin and Amin Shams – instructors at University of Texas and the Ohio State University, respectively – analyzed over 200 gigabytes of data for the transaction history between bitcoin and tether, another digital currency. Tether is an asset known as a "stablecoin," which has its trading value connected to the dollar.

The professors' study found that tethers being traded for bitcoins revealed a pattern.

https://www.cnbc.com/2019/11/04/study-single-anonymous-market-manipulator-pushed-bitcoin-to-20000.html

Where are we heading?

- **There are currently ~2300 cryptocurrencies**
- **Some are tied to real currency**
  - E.g., Tether's stablecoin backed by $
  - Designed to park funds during times of high volatility
  - But they admitted that it only has 74% in of Tether backed by cash reserves

GIZMODO

French Students Will Now Have to Learn About Bitcoin

Jennings Brown • November 1, 2019

High school students in France may be among the first people in the world to actually understand how cryptocurrency works.

The Next Web reports that the French education ministry, Le Ministère de l’Éducation Nationale, will integrate cryptocurrency into its curriculum and teach students the influence that bitcoin has on the economy. An outline of the curriculum notes that under this new module, high school teachers will provide a basic overview of cryptocurrency so students can understand the framework of decentralized financial systems.

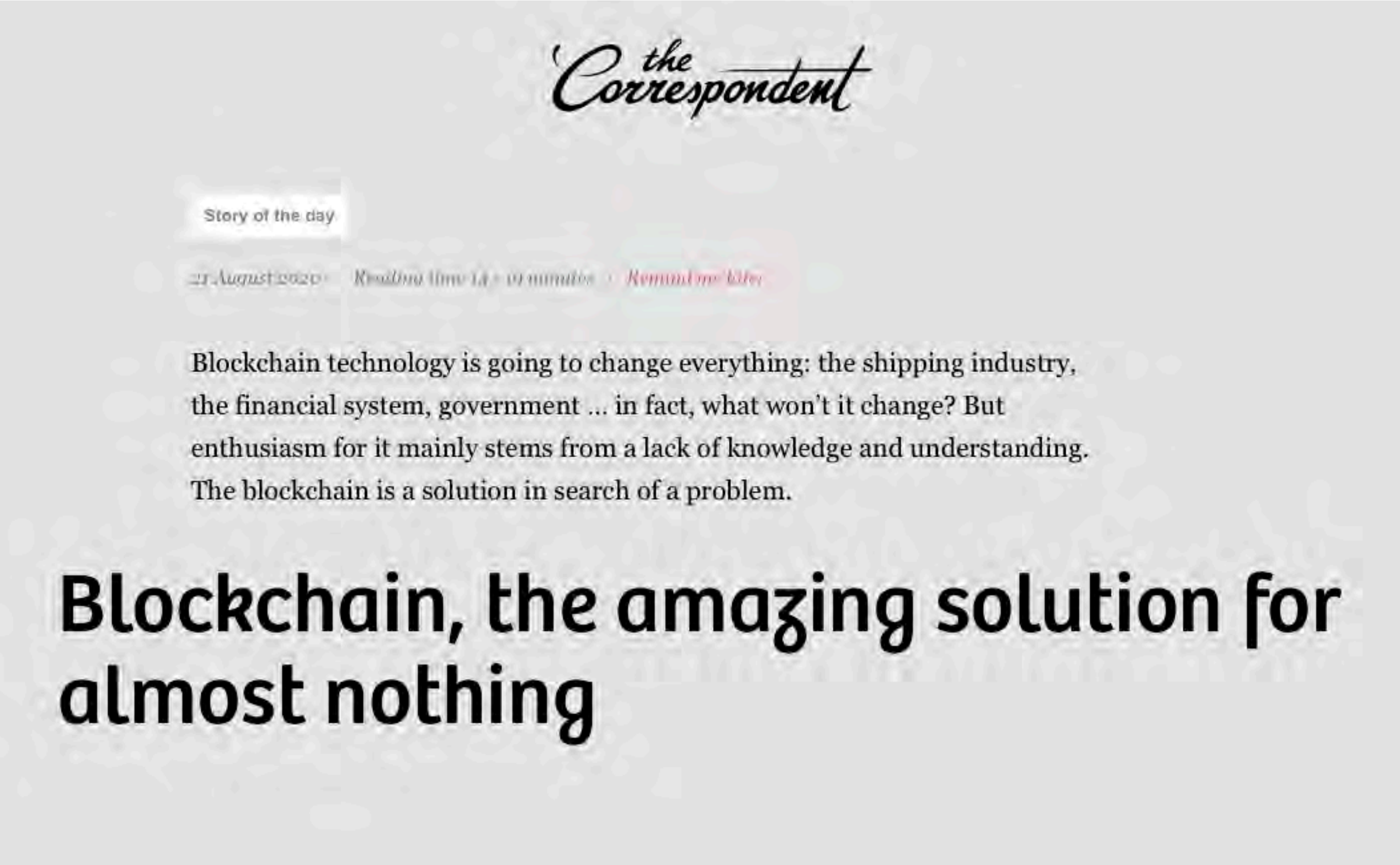
https://gizmodo.com/french-students-will-now-have-to-learn-about-bitcoin-1839547502

GREGORY BARBER BUSINESS 10.20.2019 07:00 AM

What's Blockchain Actually Good for, Anyway? For Now, Not Much

Not long ago, blockchain technology was touted as a way to track tuna, bypass banks, and preserve property records. Reality has proved a much tougher challenge.

ILLUSTRATION: ARIEL DAVIS



the Correspondent

Story of the day

21 August 2020 · Reading time 14 – 19 minutes · Remind me later

Blockchain technology is going to change everything: the shipping industry, the financial system, government … in fact, what won’t it change? But enthusiasm for it mainly stems from a lack of knowledge and understanding. The blockchain is a solution in search of a problem.

Blockchain, the amazing solution for almost nothing

https://thecorrespondent.com/655/blockchain-the-amazing-solution-for-almost-nothing/86714927310-8f431cae

The End