



ONLINE

Cryptocurrency and Blockchain: an Introduction to Digital Currencies

Introduction to Module 2

Jessica Wachter, Professor of Financial Management, Professor of Finance

Graveyard of Failed Online Currencies

- CyberCash
 - Launched in 1990's that pioneered the CyberCoin
 - Problem: every user needed to obtain a certificate to verify their identity
 - Declared bankruptcy in 2001
 - Technology was eventually acquired by PayPal.

Graveyard of Failed Online Currencies

- DigiCash
 - Used CyberBucks
 - Clients were anonymous
 - Patented a blind-signature scheme that has some similarity to Bitcoin's protocol
 - Merchants were not anonymous and needed to register with a bank
 - No user-to-user transactions
 - DigiCash declared bankruptcy in 1998
 - Technology eventually acquired by InfoSpace

Graveyard of Failed Online Currencies

- Magic Money
 - Created by a members of a mailing list called Cypherpunks
 - Violated the patent of DigiCash
 - Cypherpunks was the group out of which Satoshi emerged

Graveyard of Failed Online Currencies

- Lucre
 - DigiCash but without the patented technology

“A lot of people automatically dismiss e-currency as a lost cause because of all the companies that failed since the 1990’s. I hope it’s obvious it was the centrally controlled nature of those systems that doomed them.”

— SATOSHI NAKAMOTO, BITCOIN DEVELOPER
FEBRUARY 15TH, 2009

Problems a Decentralized Currency Must Solve

1. Must establish a system of ownership rights
 - Must be self-enforcing because it is not possible to rely on the state
2. Must maintain a ledger of transactions that is secure and accurate
 - Must also be self-enforcing because there is no profit-making intermediary behind the scenes
3. Must maintain trading in the currency
4. Must have rules governing the supply of the currency

In this Module

- The digital signature
- A tamper-proof ledger
- What is Blockchain?
- Examples of Blockchain (and why we need to go a step further)
- Creating distributed consensus
- Currency supply and trade
- Future challenges



ONLINE

Cryptocurrency and Blockchain: an Introduction to Digital Currencies

The Digital Signature

Jessica Wachter, Professor of Financial Management, Professor of Finance

The Digital Signature

Fees for Sending \$3000.00 to Turkey

Destination:
Turkey

Receive Currency:
TRY

ZIP Code
19066

Amount⁴
3000.00 USD

Update »



Send online

Send in person

Use mobile app

Send

For cash pickup at an agent location

	PAY WITH Bank account	SERVICE¹ 4 Business days	SENDING 3000.00 USD	FEE 6.00 USD	Send
Estimated Exchange Rate: 1 USD = 5.0649 TRY ³					
	PAY WITH Credit/debit card ²	SERVICE¹ In Minutes	SENDING 3000.00 USD	FEE 46.00 USD	Send
Estimated Exchange Rate: 1 USD = 5.0649 TRY ³					

¹ Funds may be delayed or services unavailable based on certain factors, including the amount sent, destination country, currency availability, regulatory requirements, agent location hours, differences in time zones, and selection of delayed delivery options. Additional restrictions may apply; see our terms and conditions.

The Digital Signature

Fees for Sending \$3000.00 to Turkey



Destination:
Turkey ▼

Receive Currency:
TRY ▼

ZIP Code
19066

Amount⁴
3000.00 USD

Update »

Send online	Send in person	Use mobile app	Send
For cash pickup at an agent location			
 PAY WITH Bank account	SERVICE ¹ 4 Business days	SENDING 3000.00 USD	FEE 6.00 USD
Estimated Exchange Rate: 1 USD = 5.0649 TRY ³			
 PAY WITH Credit/debit card ²	SERVICE ¹ In Minutes	SENDING 3000.00 USD	FEE 46.00 USD
Estimated Exchange Rate: 1 USD = 5.0649 TRY ³			

¹ Funds may be delayed or services unavailable based on certain factors, including the amount sent, destination country, currency availability, regulatory requirements, agent location hours, differences in time zones, and selection of delayed delivery options. Additional restrictions may apply; see our terms and conditions.

The Digital Signature

Fees for Sending \$3000.00 to Turkey

Destination:
Turkey

Receive Currency:
TRY

ZIP Code
19066

Amount⁴
3000.00 USD

Update »



Send online

Send in person

Use mobile app

Send

For cash pickup at an agent location

	PAY WITH Bank account	SERVICE¹ 4 Business days	SENDING 3000.00 USD	FEE 6.00 USD	Send
Estimated Exchange Rate: 1 USD = 5.0649 TRY ³					
	PAY WITH Credit/debit card ²	SERVICE¹ In Minutes	SENDING 3000.00 USD	FEE 46.00 USD	Send
Estimated Exchange Rate: 1 USD = 5.0649 TRY ³					

¹ Funds may be delayed or services unavailable based on certain factors, including the amount sent, destination country, currency availability, regulatory requirements, agent location hours, differences in time zones, and selection of delayed delivery options. Additional restrictions may apply; see our terms and conditions.

Property Rights in Bitcoin

- Cryptocurrency must create its own system of property rights
 - Must be self-enforcing, because the system is decentralized

Property Rights in Bitcoin

- Who possess rights within the system, while still maintaining anonymity?
- How do you prevent this from being so cumbersome that individuals do not want to join? (recall DigiCash...)

Property Rights in Bitcoin

- Bitcoin has an elegant answer!
 - It dispenses with the notion of the human being in the background entirely
 - You “are” your signature
 - You can have as many signatures as you want
 - Your signature has a private component (the private key), and the public component
 - The public signature is synonymous with the address

The Digital Signature

- What is a signature?
 - An object with the following properties:
 1. Only you can make it (unforgeability)
 2. Anyone can verify it
 3. It's permanent

The Digital Signature

- A digital signature:
 1. A private key, generated at random
 2. A protocol for affixing the private key to an electronic message (this is the actual written signature)
 3. A protocol for verifying that your signature is valid
 - Without revealing your private key
 - This is where the public part of your signature comes in

The Digital Signature

- It is very important that:
 - no one can forge your signature
 - no one can replicate your signature after reading a message
- Thus the creation of the signature must be random
- Affixing the signature to the message must be encrypted, which also involves randomness

The Digital Signature

- Think about what makes physical signatures so difficult to forge
 - Created by a human being
 - A built-in randomness to being human: no two humans are alike
 - No person is the same every day. One cannot replicate a signature just by seeing it.
 - Thus humanity interposes randomness at both steps of the process

The Digital Signature

- “Random-number generators”
 - Software that produces a string of random numbers
 - Computer-generated random numbers are “pseudo-random numbers”

Summary

- Bitcoin is a decentralized system of property rights
- Key to the notion of property rights is a notion of identity
- To verify identity is through the use of a signature
- Bitcoin makes these one and the same
- A digital signature defines each Bitcoin transaction
- These signatures, at the core of Bitcoin, require randomness and thus are imperfect
- So far, this imperfection has not proven to be problematic.



ONLINE

A Tamper Proof Ledger

Jessica Wachter, Professor of Financial Management, Professor of Finance

The Ledger

- The ledger contains a record of all Bitcoin transactions
 - A memory system
- The equivalence between money and memory was proposed by economist Narayana Kocherlakota in 1998
- Along with the digital signature, the accurate ledger is required for the existence of property rights in Bitcoin

Bitcoin Innovation: Incentives

- Satoshi's solution
 - In previous attempts to decentralize, creators focused on making it impossible to tamper with the ledger
 - Satoshi realized that it was sufficient (and much easier) to have incentives not to tamper with the ledger

Bitcoin Innovation: Incentives

- How to dis-incentivize participants from tampering with the ledger?
- Answer: By making it easy to detect that the ledger had been tampered with
 - Any dishonest participant would then be dissuaded from even trying
- Tampering with the ledger is what, in economics, we call an “off-the-equilibrium path.”
 - No one will do it, but incentives to prevent it exists

Tampering with the Ledger

- Consider a simple example
 1. Maria creates a single MariaCoin
 2. Maria sells the MariaCoin to Sophie
 3. Sophie sells the MariaCoin to Geoff
- Thus it is clear that, at the end, Geoff is the rightful owner of MariaCoin
- But what if Bob were to come in and attempt to change the second stage: to read that Maria sells MariaCoin to Bob, not Sophie.
 - This would undermine Geoff's ownership rights to the coin
- We need to detect possible tampering all along the chain

Detecting Tampering Attempts

- Solution: We make the ledger recursive:
 1. Maria creates a single MariaCoin
 2. Maria sells the MariaCoin to Sophie <MCSM>
 3. Sophie sells the MariaCoin to Geoff <MSMS<mcsm>>
 4. <SSMG <msms<mcsm>>>>
- Alongside each entry, I have put a “digest” – a kind of shorthand – to the previous entry
- In creating this shorthand, I followed a simple algorithm. I used the first letter of each of the words.
- To detect tampering, just check if the initials match the words in the previous step

Detecting Tampering Attempts

1. Maria creates a single MariaCoin
2. Maria sells the MariaCoin to Sophie <MCSSM>
3. Sophie sells the MariaCoin to Geoff <MSMS<mcssm>>
4. <SSMG <msms<mcssm>>>>

Detecting Tampering Attempts

1. Maria creates a single MariaCoin
2. Maria sells the MariaCoin to Bob <MCSM>
3. Sophie sells the MariaCoin to Geoff <MSMS<mcsm>>
4. <SSMG <msms<mcsm>>>>

Detecting Tampering Attempts

1. Maria creates a single MariaCoin
2. Maria sells the MariaCoin to Bob <MCSM>
3. Sophie sells the MariaCoin to Geoff <MSMS<mcsm>>
4. <SSMG <msms<mcsm>>>>

Detecting Tampering Attempts

1. Maria creates a single MariaCoin
2. Maria sells the MariaCoin to Bob <MCSM>
3. Sophie sells the MariaCoin to Geoff <MSMB<mcsm>>
4. <SSMG <msms<mcsm>>>>

Detecting Tampering Attempts

1. Maria creates a single MariaCoin
2. Maria sells the MariaCoin to Bob <MCSM>
3. Sophie sells the MariaCoin to Geoff <MSMB<mcsm>>
4. <SSMG <msms<mcsm>>>>

Detecting Tampering Attempts

1. Maria creates a single MariaCoin
2. Maria sells the MariaCoin to **Bob** <MCSM>
3. Sophie sells the MariaCoin to Geoff <MSMB <mcsm>>
4. <SSMG <msms <mcsm>>>>

Detecting Tampering Attempts

1. Maria creates a single MariaCoin
2. Maria sells the MariaCoin to Bob <MCSM>
3. Sophie sells the MariaCoin to Geoff <MSMB<mcsm>>
4. <SSMG <msms<mcsm>>>>

Detecting Tampering Attempts

1. Maria creates a single MariaCoin
2. Maria sells the MariaCoin to Bob <MCSM>
3. Sophie sells the MariaCoin to Geoff <MSMB<mcsm>>
4. <BSMG <msmb<mcsm>>>>

Summary

- A tamper-proof ledger is a key feature of Bitcoin that enforces property rights
- How do you make a ledger tamper-proof when it is distributed?
 - You make sure any attempts will be discovered
 - By making the ledger recursive. Every entry contains a little copy of the previous entry.



ONLINE

Cryptocurrency and Blockchain: an Introduction to Digital Currencies

What is Blockchain?

Jessica Wachter, Professor of Financial Management, Professor of Finance

What is Blockchain?

- Blockchain is the same idea but with cryptography so that it is hard to work around
- Blockchains store large groups of transactions into blocks, not just one at a time

What is Blockchain?

- The linked list
- The hash function
- The hash pointer
- Putting it together: Blockchain
- How blockchain creates a truly tamper-proof ledger

What is Blockchain?

- Pointer
 - A language object that stores the memory address of another value located in computer memory
- Linked list
 - A linear collection of data elements such that each element contains a pointer that points to the next
 - The elements might be in entirely different places, but they are connected by pointers
 - The pointers turn a collection of objects into an ordered list
 - Could be a list of financial transactions → a ledger!

The Hash Function

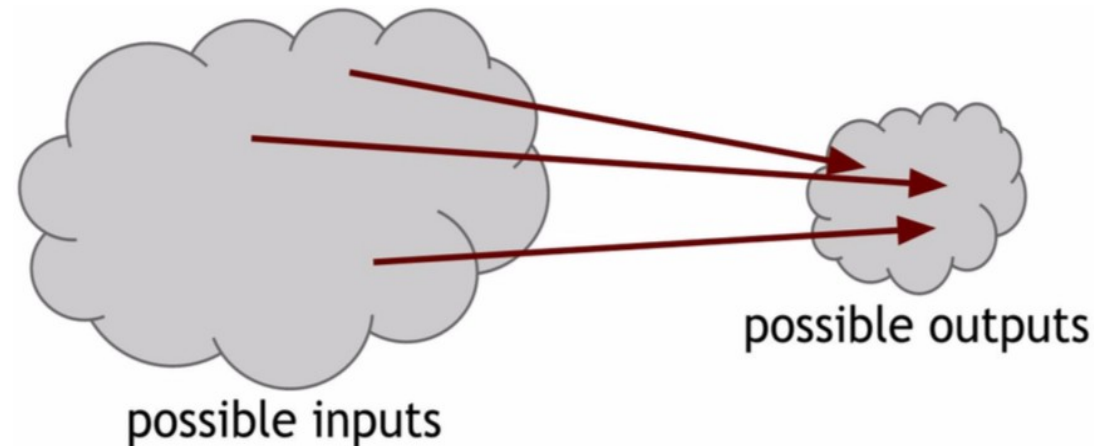
1. Maria creates a single MariaCoin
 2. Maria sells the MariaCoin to Sophie
 3. Sophie sells the MariaCoin to Geoff
- We would digitally implement this list by turning each description into an object, and creating a link to the previous object.

A Quick Summary So Far

- A pointer indexes an object in computer memory
- Using a system of objects connected by pointers, one can represent a ledger
- How we make that ledger tamper-proof?
 - We replace the pointer with a hash pointer
 - The hash pointers turn the linked list into a block chain

The Hash Function

- A function takes an input and returns an output
 - Can't have more than one output per input
 - Can have more than one input per output



- Hash function – takes an input of (virtually) any size and returns an output of a fixed size
 - In this case a 256-bit number

The Hash Function

1. Maria creates a single MariaCoin
2. Maria sells the MariaCoin to Sophie <MCSM>
3. Sophie sells the MariaCoin to Geoff <MSMS<mcsm>>
4. <SSMG <msms<mcsm>>>>

The Hash Function

- Replace the digest method with one using hash functions
 - SHA-256 - Same has function as Bitcoin
 - Takes anything and turns it into a 256-bit number
 - <https://www.movable-type.co.uk/scripts/sha256.html>



Movable Type Scripts

SHA-256 Cryptographic Hash Algorithm

A **cryptographic hash** (sometimes called 'digest') is a kind of 'signature' for a text or a data file. SHA-256 generates an almost-unique 256-bit (32-byte) signature for a text. See **below** for the source code.

Enter any message to check its SHA-256 hash

Message

Hash

e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855

1.410ms

Note SHA-256 hash of 'abc' should be: ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad

The Hash Function

The ledger with hashes of the previous entry:

1. Maria creates a single MariaCoin
2. Maria sells the MariaCoin to Sophie
8a6daa572851240bb8bf3268e0083e02ed6008ebd8aaf3d46a47baa0738afcae
3. Sophie sells the MariaCoin to Geoff
9e324bbd1e6bfb84ef216cd1022b0578d80dfe831d3cb0939c74b8fad57c29b6
4.
5fb54613a78f5fdfe85f3f6a592ab2f4e0025896d69e3512ed1692a46f9392c3

The Hash Function

- SHA-256 is a cryptographic hash function
 - Two inputs are very unlikely to produce the same output (*collision-resistance*)
 - It must be possible to find inputs that produce the same output
 - The input space is much bigger than the output space
 - For SHA-256, the chance that any two random inputs would have the same output is 2^{256}
 - To put this in perspective, there are an estimated 2^{259} atoms in the universe

The Hash Function

- SHA-256 is a cryptographic hash function
 - It satisfies Collision-Resistance
 - Hiding: given an output it is virtually impossible to reverse-engineer the input
 - Puzzle-friendliness: used to make good puzzles!

The Ledger with a Hash Digest

- Even a tiny amount of tampering will produce inconsistency between the statement and the hash
- Collision-resistance and hiding means that one cannot cleverly tamper to produce the same hash
- Recursivity means one has to tamper all along the chain to not be discovered

Block Chain

- It is a linked-list
 - With a hash pointer instead of a pointer linking objects
- The objects are not individual transactions, but rather blocks with several thousand transactions
- A hash pointer is the hash function applied to the previous entry when it was created

Summary

- To implement property rights within Bitcoin, we require a tamper-proof ledger
- Begin with a linked list (a ledger)
- Replacing the pointer with a hash pointer makes the ledger tamper-proof
 - It turns it into a blockchain
 - Technically, blockchain is not tamper proof
 - It is tamper-proof in the strongest sense: no-one has an incentive to try



ONLINE

Cryptocurrency and Blockchain: an Introduction to Digital Currencies

Examples

Jessica Wachter, Professor of Financial Management, Professor of Finance

Back to MariaCoin

1. Maria creates MariaCoin
 - She creates a unique digital ID representing the coin and signs it with her private key
2. Maria writes a message “Pay Sophie with this coin.”
 - Sophie = Sophie’s public signature
 - This coin = a hash pointer to the coin
3. Maria signs the message with her private key
 - Using Maria’s public key, any user can view this series of operations and verify that Sophie is the valid owner of the coin
 - The ledger is secure, property rights are valid
 - Sophie can repeat steps 2 and 3 to transfer the coin where she likes

The Double-spend Attack

Sophie can repeat the protocol to pay another user (Geoff) with the coin

AND

Can then repeat the protocol again to pay yet another user, (Mike) with the coin

- There are no safeguards in the system to prevent Sophie from spending the coin twice

Avoiding the Double-spend Attack

- JamesCoin
 - Like MariaCoin, but with one key difference
 - All transactions in the ledger must be signed by James to be valid
1. James creates JamesCoin
 2. James writes a message “Pay Sophie with this coin.”
 3. Sophie can then write a message “Pay Geoff with this coin.”
 4. Both Sophie *and James* sign off on this last message
 - Because James has signed the message “Pay Geoff with this coin” his software “knows” that Geoff is the valid owner
 - It will not let him sign off on a contradictory message

Avoiding the Double-spend Attack

- The ledger can be widely distributed, and is tamper-proof, due to blockchain technology
- Even James himself cannot manipulate the ledger without detection
- For these reasons, this is more decentralized than a credit card or bank

Avoiding the Double-spend Attack

- Problems with JamesCoin
 - He can sign off on his friends' transactions and not on others
 - If something happens to James, there is no longer a currency
 - Bitcoin seeks a more decentralized system

Summary

- Blockchain and the digital signature are necessary components for cryptocurrency
 - But they are not sufficient
 - They do not prevent the double-spend attack
- Requiring one address to sign off on all transactions does avoid the double-spend attack
 - Prone to manipulation and system failure
 - This is why cryptocurrency requires distributed consensus



ONLINE

Cryptocurrency and Blockchain: an Introduction to Digital Currencies

Distributed Consensus

Jessica Wachter, Professor of Financial Management, Professor of Finance

Distributed Consensus: What is it?

- A distributed network of systems running Bitcoin software
 - Consists of thousands of nodes (computers)
 - Each node is connected to every other node in the network
 - The system has no (built-in) central authority
 - It is fully peer-to-peer

Distributed Consensus

- Goal – to create a protocol that allows all nodes to agree
 - Not sufficient
 - We want the nodes to agree on “the truth.”
 - Some nodes can be honest, some can be malicious

Distributed Consensus

- Distributed consensus protocol
 - A network of nodes, each receiving a value
 - Some are malicious, some honest
 - Must terminate with all honest nodes in agreement on the value
 - The value must have been generated by an honest node

Application to the Blockchain

- Assume that the system starts off in a good state
 - All honest nodes are in agreement on the ledger
- New transactions come in
 - Some potentially contradictory
- The distributed consensus protocol will end with a new block added to the ledger, consisting of valid transactions, on which all nodes agree

How Do We Get There?

- Bitcoin protocol avoids assuming that any one node is honest
 - Avoids assigning stable identities to the nodes at all
- We cannot prefer any one node over the other
- How to reach consensus
 - Use randomness¹

¹This thought experiment is adapted from *Bitcoin and crypto technologies*.

Brief Review of the Hash Pointer

- A hash pointer to a block is:
 - the output of the hash function applied to that block (“the hash”)
 - and a pointer to that block
- Each block in the blockchain includes a hash pointer to the previous block

Bitcoin Distributed Consensus Algorithm – First Pass

1. New transactions are broadcast to all nodes
2. Each node collects these transactions into a block
3. At a fixed interval, a *random* node gets to propose its block
 - Including a hash pointer to this previous block
4. All nodes check the block to make sure the transactions are valid

Repeat steps 1 - 4

Distributed Consensus – First Pass

- This method avoids some obvious problems
 - Not possible to spend bitcoins not belonging to you
 - Not possible to deny service to a certain user whom you don't like
 - That user just waits for the turn of an honest node
 - It also avoids some more subtle problems

Avoiding the Double-spend Attack

- MariaCoin



Avoiding the Double-spend Attack

- MariaCoin



Avoiding the Double-spend Attack

- MariaCoin



Avoiding the Double-spend Attack

- MariaCoin



- Honest nodes should include the message first
- Some nodes may be dishonest nodes and be bias
- As long as the nodes are running the software, they won't be able to include both

Avoiding the Double-spend Attack

- MariaCoin



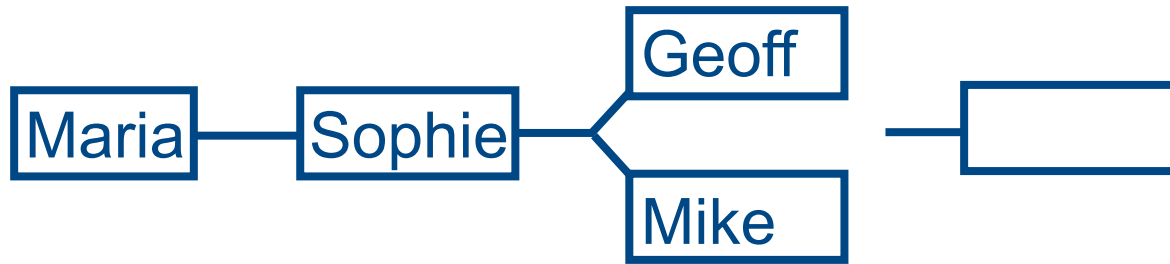
Avoiding the Double-spend Attack

- MariaCoin



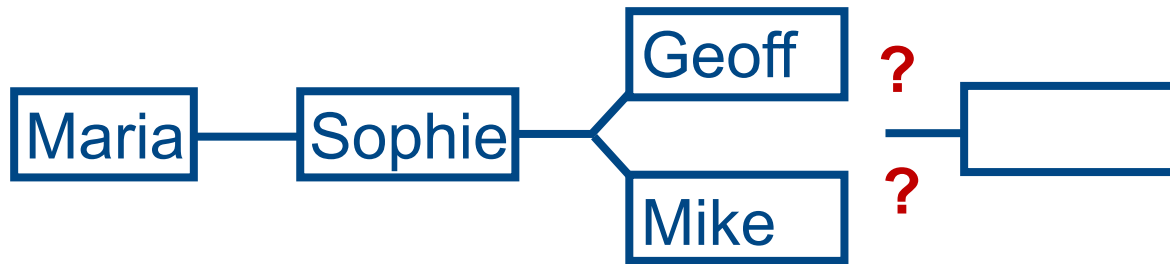
Avoiding the Double-spend Attack

- MariaCoin



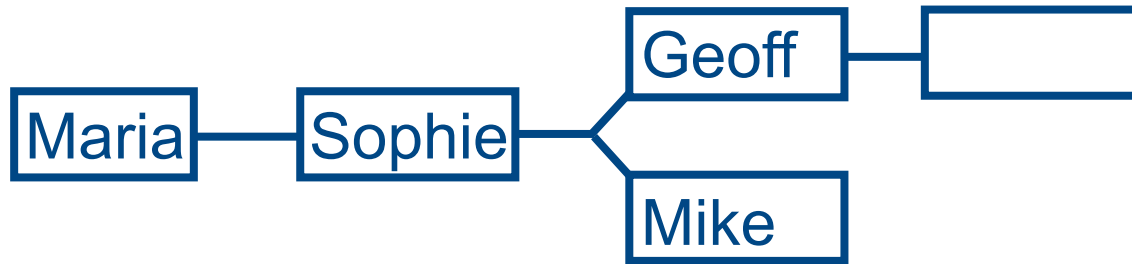
Avoiding the Double-spend Attack

- MariaCoin



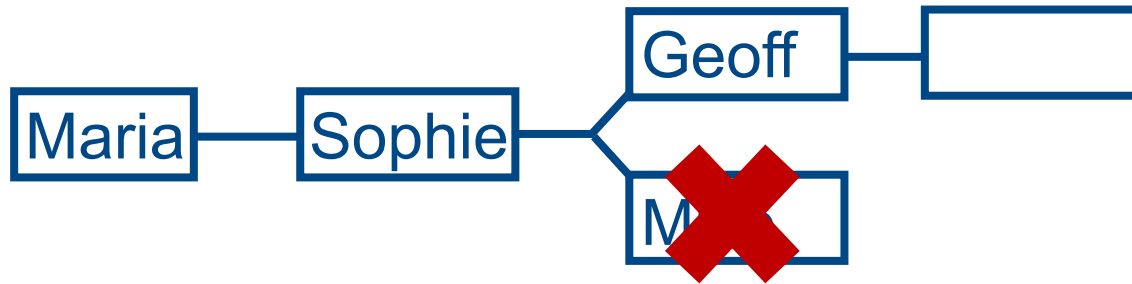
Avoiding the Double-spend Attack

- MariaCoin



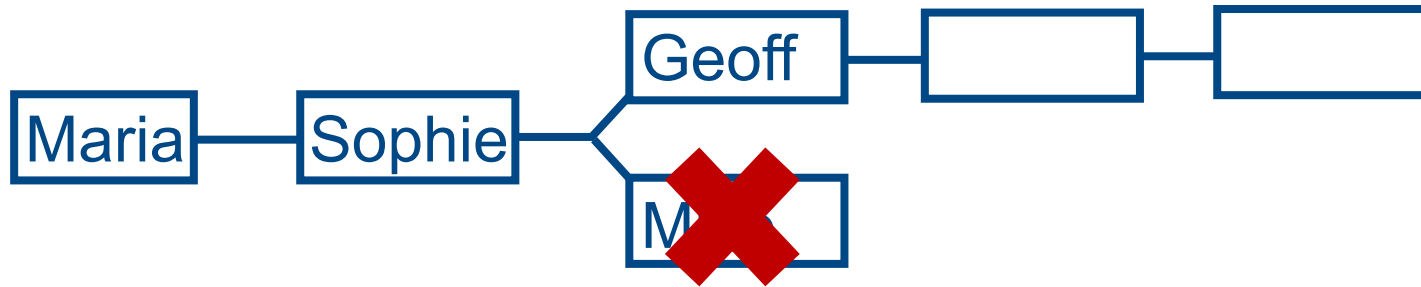
Avoiding the Double-spend Attack

- MariaCoin



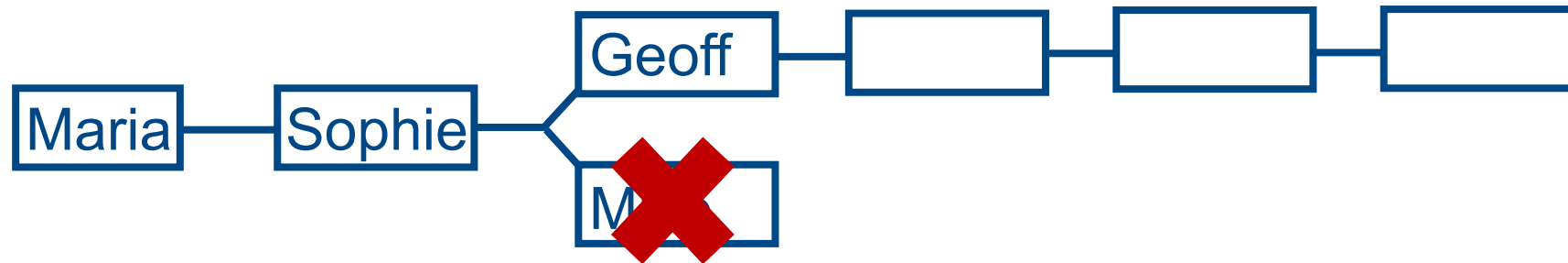
Avoiding the Double-spend Attack

- MariaCoin



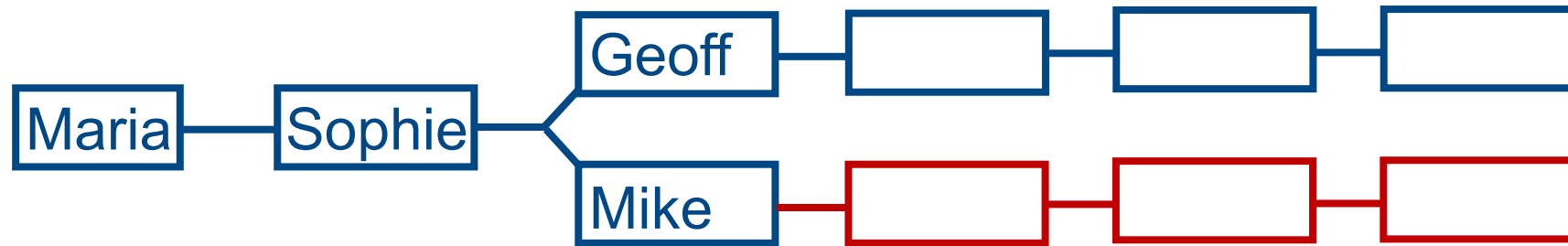
Avoiding the Double-spend Attack

- MariaCoin



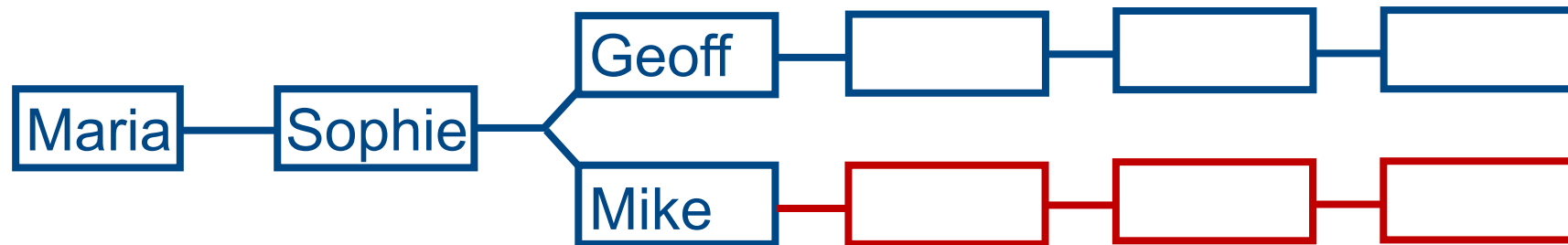
Avoiding the Double-spend Attack

- MariaCoin



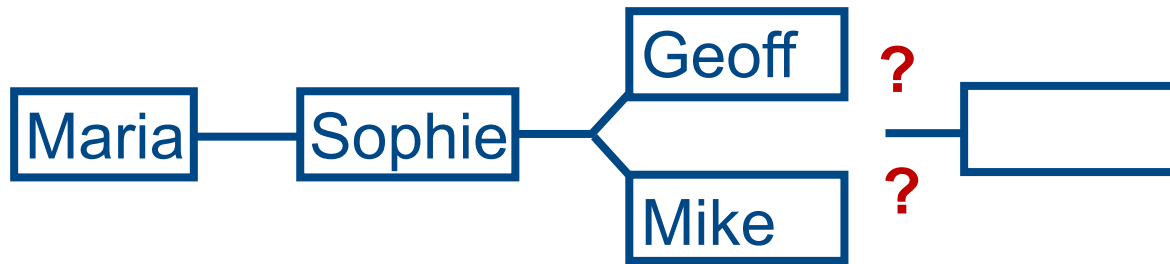
Consequence: Transaction Latency

- MariaCoin



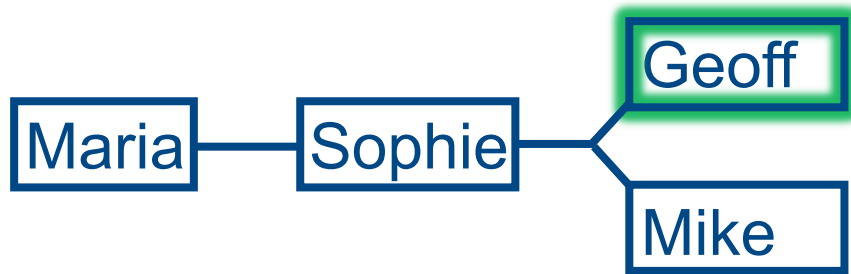
Consequence: Transaction Latency

- MariaCoin



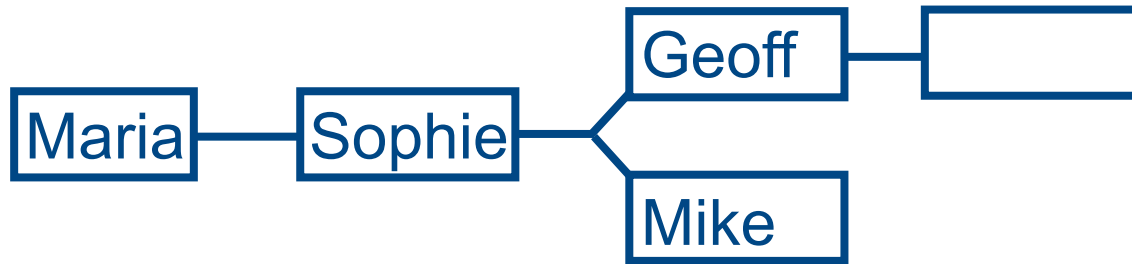
Consequence: Transaction Latency

- MariaCoin



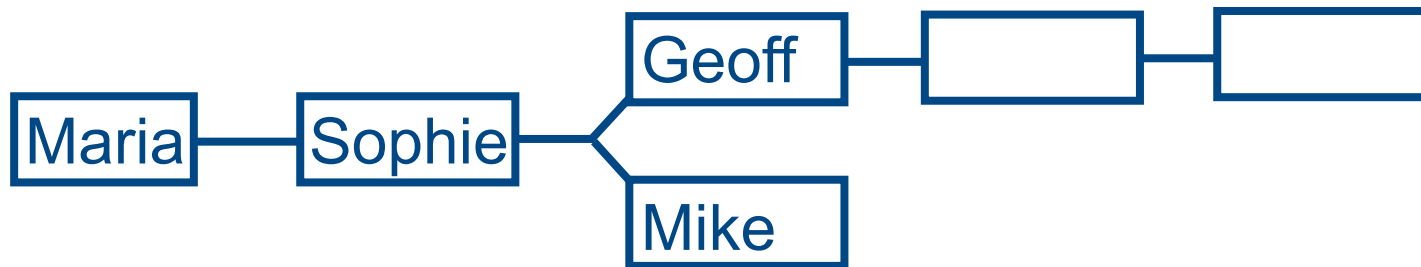
Consequence: Transaction Latency

- MariaCoin



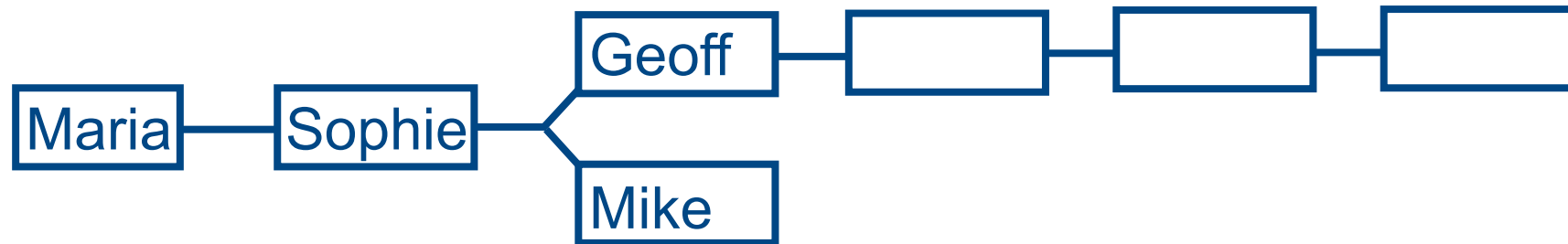
Consequence: Transaction Latency

- MariaCoin



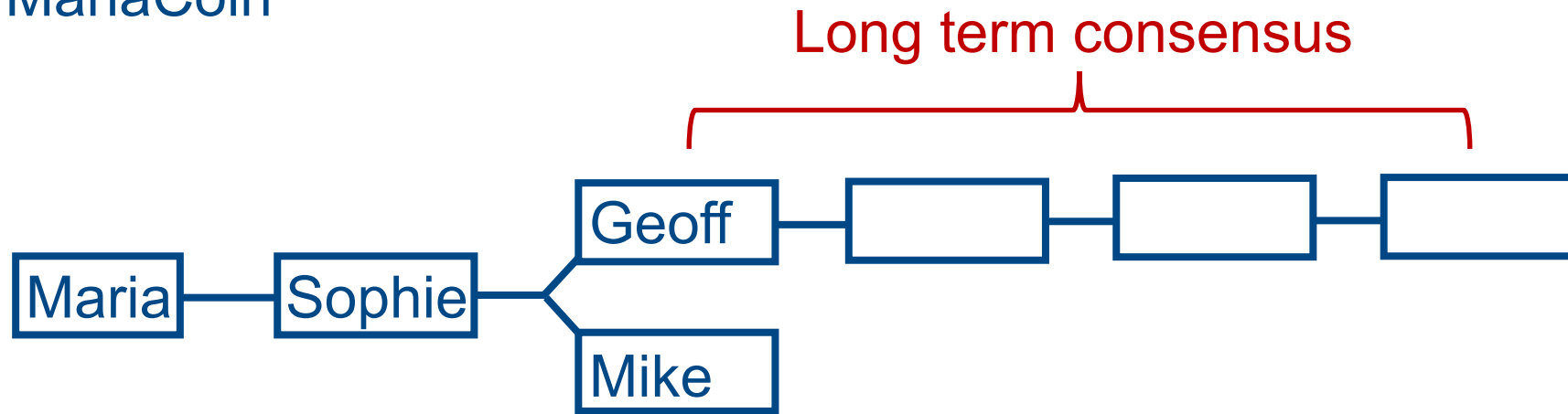
Consequence: Transaction Latency

- MariaCoin



Consequence: Transaction Latency

- MariaCoin



Summary

- First pass through Bitcoin's distributed consensus protocol
 - Transactions are broadcast to the network
 - Nodes collect the transactions into a block
 - Nodes are chosen at random to propose a block
 - In subsequent rounds, other nodes accept or reject that block
 - Disagreements become forks in the blockchain
 - Short forks are usually abandoned – it is the policy of honest nodes to extend the longest fork

Summary

- Eventually, the system reaches consensus on one chain, with near certain probability
- There is always some probability that an abandoned fork could be picked up again
- Property rights in Bitcoin are inherently probabilistic
 - This may make them more stable than the sovereign-dependent rights we are used to



ONLINE

Cryptocurrency and Blockchain: an Introduction to Digital Currencies

Proof of Work

Jessica Wachter, Professor of Financial Management, Professor of Finance

The Problem with Distributed Consensus

- Distributed consensus protocol
 - Pick nodes at random
 - Makes it hard for any single node to control the blockchain
 - Automatically disincentivizes malicious behavior

Proof of Work

- Ideally like to do is only pick honest nodes, and never pick malicious ones
 - This is impossible
 - Nodes do not have identities
- What if you could somehow turn a node into an honest node?
 - Impossible for two reasons
 - There is no such thing as computer-generated randomness
 - It would require a centralized computer to generate the randomness, and everyone would need to agree on it

Proof of Work

- Two problems to solve:
 1. Generating randomness
 2. Incentivizing honest behavior
- Solving one problem helps with the other:
 - If randomness is not good enough, it makes it more likely a malicious node will take over the system
 - If we incentivize honest behavior, then fewer nodes are malicious
 - Decreasing the need for true randomness

Proof of Work

- It's worth understanding PoW for at least two reasons:
 1. Contains some genuinely new ideas
 - Many intermediaries profit off of information (Uber, Facebook, etc.)
 - What if information could be stored and accurately maintained in a decentralized way?
 2. On the other hand: many believe that PoW contains the seeds of Bitcoin's downfall

PoW Concept 1: Block Reward

- Pick a node at random to propose a block
- Give the node that proposes the block some extra bitcoins to transact with
- As part of the block, it gets to include a transaction with these bitcoins (presumably to pay itself)
- This acts to keep blocks honest

PoW Concept 1: Block Reward

- How does block reward act to keep a node honest?
 - The only way to get to receive your reward is if future nodes accept their block
 - They do this by including a hash pointer to your block in the next block they propose
 - If you act maliciously, extending, an orphaned fork with transactions to you or your friends
 - The next node is unlikely to accept your block

PoW Concept 1: Block Reward

- Playing devil's advocate
 - Suppose most nodes are not honest, but rather are malicious
 - If you are malicious, the next node might choose to reward your behavior by accepting your block
 - If you are honest, the next node might choose to punish you by accepting a different block
 - Then dishonest behavior would be rewarded, and that would be bad

PoW Concept 1: Block Reward

- Rely on the fact that nodes behave honestly unless incentivized otherwise
- Malicious nodes are out for their own good
- Not assuming that:
 - malicious nodes are designed for the destruction of the system

PoW Concept 1: Block Reward

- Why is this?
 - I, Jessica, propose a block that exhibits self-dealing
 - The next node chosen to propose a block (Alice)
 - Would Alice accept my block, rewarding my malicious behavior?
 - Alice also wants to receive the block reward!
 - Accepting my block makes it less likely that *her* block will be accepted in the next round
 - She needs to be both malicious and without self-interest
 - In the background: a prevailing view that most nodes are acting honestly, because all incentives are for them to do so

Summary

- In the Bitcoin protocol, all nodes are “peers”
- They all get to propose transactions
- Why should they behave honestly, as opposed to seek to reward themselves
 - Answer: incentives!
- Honest behavior leads to (likely) block reward
- Malicious behavior probably will simply not succeed



ONLINE

Cryptocurrency and Blockchain: an Introduction to Digital Currencies

Mining and Currency Supply

Jessica Wachter, Professor of Financial Management, Professor of Finance

Bitcoin Mining

- Bitcoin mining is a resource-intensive activity that results in the discovery of new coins
- Thus the analogy to precious metals implicit in the word “mining”

The Hash Puzzle

- Nodes compete to have a chance to propose the next block
- They succeed if they are the first to solve a hash puzzle
- The hash puzzle is a cryptographic puzzle
- Puzzle-friendliness implies that these puzzles can be found
- The only way to solve the cryptographic puzzle is through trial and error

The Hash Puzzle

- The greater the computing power of the node, the more likely it is to solve the hash puzzle
- When it solves the hash puzzle, it proposes the block, with the potential to receive the block reward (if the block is accepted)
- It has thus “mined” new bitcoins
- Important: all nodes can verify that a given node has solved the puzzle

Proof of Work and Randomness

- Proof of work is therefore the solving of a hash puzzle
- For which the node is rewarded by being able to propose a block
- And for the block reward, given in bitcoins
- Notice that proof of work creates the randomness that we were missing!

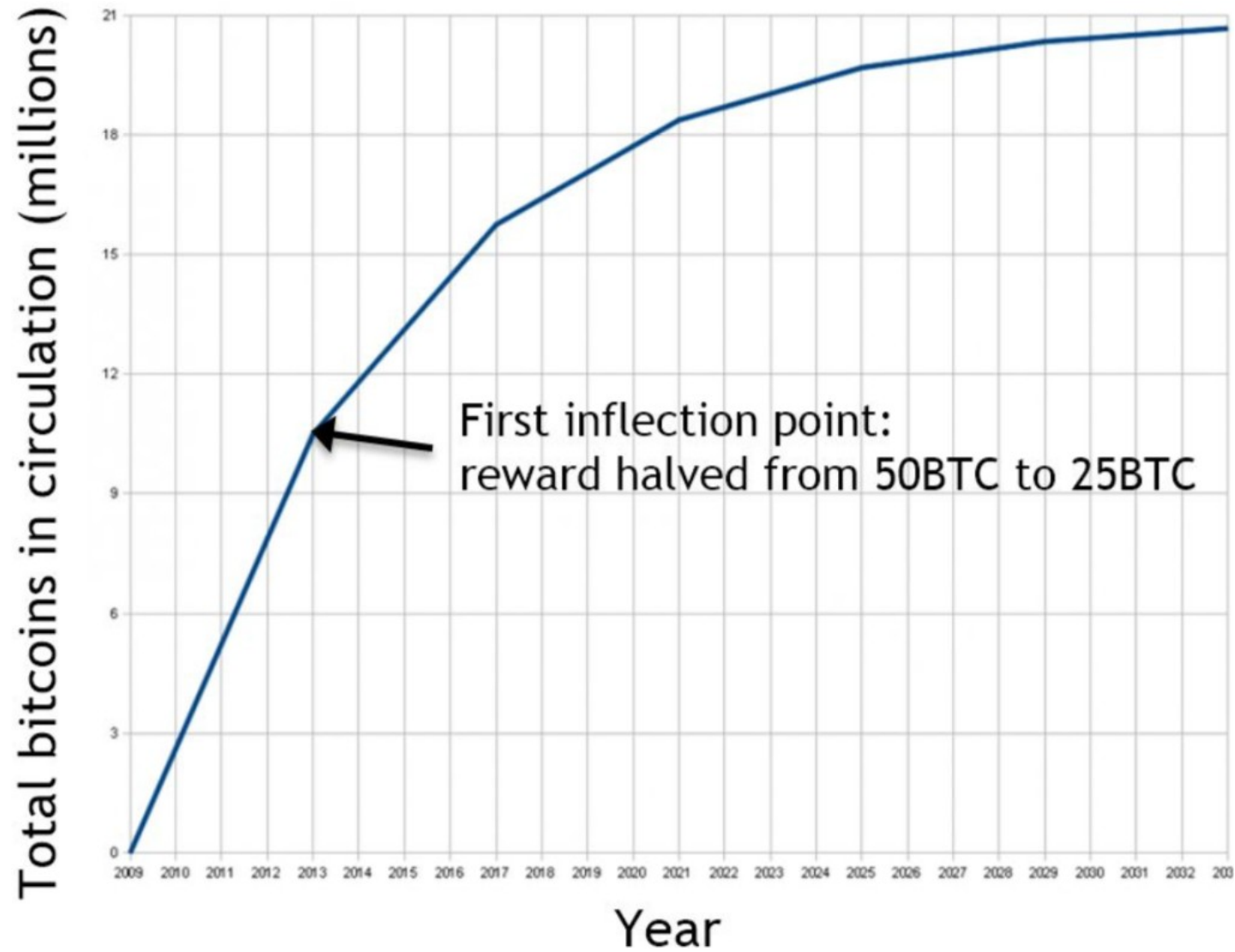
Proof of Work and Randomness

- For two competing nodes with roughly equal processing power, there is no way to predict which node will solve the puzzle first and get to propose its block
- Is block selection random?
 - Unpredictable? Yes.
 - Only way to predict which one will win is by having as much computing power as the two nodes combined
 - Nondeterministic? No.
 - Note: a process can be both deterministic and random (Henri Poincare in the 19th century)

Currency Creation

- Block rewards
 - Only way within the Bitcoin reference software for new coins to be mined
 - Even block rewards are limited
- Every 210,000 blocks, the block reward is cut in half
 - Occurs every 4 years (approximately)
 - Implies the total number of bitcoins will converge to 21 million

Total Bitcoins in Circulation



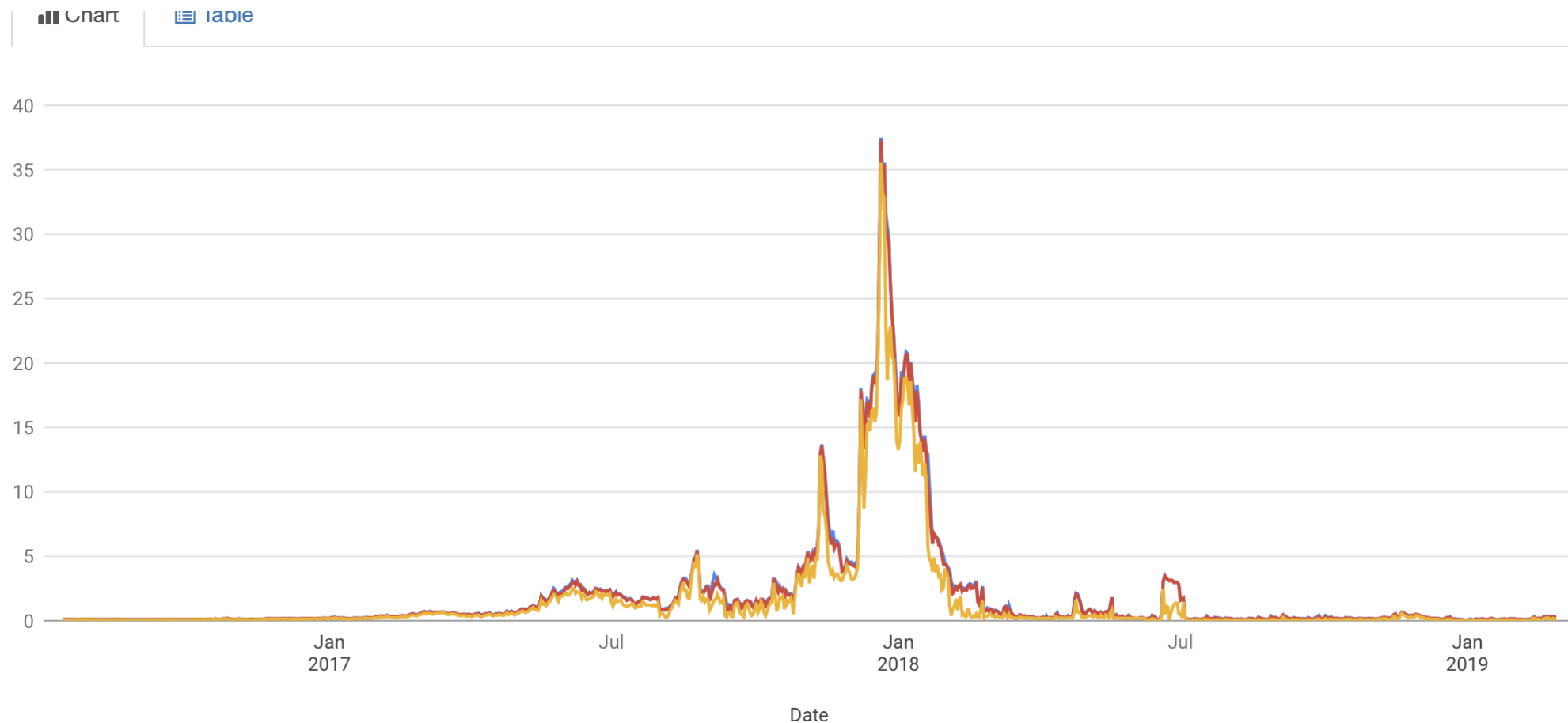
Currency Creation

- The creation of new bitcoins allows the currency to be in limited supply at the beginning and the supply to grow and then to converge
- It does *not* imply that Bitcoin is subject to deflationary pressures
- It does imply that Bitcoin supply cannot be manipulated by a central bank

The Transaction Fee

- The potential rewards to mining go beyond receipt of bitcoins
 - Includes the receipt of transaction fees
 - The node proposes transactions in which the value of bitcoins coming in exceed the value of bitcoins going out
 - The remainder get to be paid to an address of the node's choosing
- Ensures continuing incentives for trade in Bitcoin

Bitcoin Transaction Feed



Fees skyrocketed in late-2017, but are now back to negligible (\$0.31 per transaction at the time of this writing)

Summary

- Proof of work incentivizes behavior through block reward
 - Creates randomization by making nodes compete to be the first to solve a hash puzzle
 - Randomization only among nodes with the great computing power
- Proof of work builds in currency creation and then ensures stabilization
 - Transaction fees incentivize honest trading



ONLINE

Cryptocurrency and Blockchain: an Introduction to Digital Currencies

Future Challenges

Jessica Wachter, Professor of Financial Management, Professor of Finance

PoW Challenge 1: The 51% Attack

- What if a node succeeds in gaining a majority of the CPU power across all nodes?
 - This node would be the first to solve all the hash puzzles
 - This node could then build the longest chain in the blockchain
 - The system would then revert to being centralized

PoW Challenge 1: The 51% Attack

- If a node were to obtain a majority of CPU power it would have to act as a benevolent dictator
- The only possible reward to CPU power is bitcoins
- A malicious 51% attacker could include lots of payments to him- or herself
 - No other nodes would actually let this attacker spend the coins

PoW Challenge 1: The 51% Attack

- Any such attack is limited in its profitability
- Any such attacker has an incentive to maintain trading in bitcoins
- In 2014, a mining pool came close and voluntarily capped their control at 39.9%¹

¹Wilhelm, Alex. "Popular Bitcoin Mining Pool Promises To Restrict Its Compute Power To Prevent Feared '51%' Fiasco". TechCrunch. Archived from the original on 5 December 2017.

Related: Oligopoly Power

- A weaker version of the 51% attack is what would happen if a small number of nodes were to control all mining power
- As bitcoin mining has become industrialized, this has been occurring (see the white paper “Analysis of Large-Scale Bitcoin Mining Operations.”)
- Miners could form a cartel and charge high transaction fees
 - Could be seignorage, under another name

PoW Challenge 2: Resource Intensity

- Bitcoin mining uses lots of electricity
 - High ends of estimated amounts consumed are only 6% of the global banking sector
 - If it were to increase 100-fold, it would be about 2% of total consumption ²
 - Mining takes place where energy is not a scarce resource

²Roberts, Paul (9 March 2018). "This Is What Happens When Bitcoin Miners Take Over Your Town - Eastern Washington had cheap power and tons of space. Then the suitcases of cash started arriving". Politico

Other Concerns Regarding Energy Use

- A “Green” critique of Bitcoin
 - Miners are using energy for no purpose
- PoW is a technology that enables trading in bitcoin
 - Implemented by a zero-sum game
 - As long as one believes Bitcoin is a net benefit to society, there is a net benefit to mining

PoW Challenge 3: Can the Network Scale?

- Blocks are limited in size
- Blocks can only be created about every 10 minutes
 - Implies that there are 7 transactions per second
- Trilemma: Blockchain systems can have at most 2 of the three:
 1. Decentralization
 2. Scalability
 3. Security

Module Summary

- Bitcoin creates a system of property rights that does not depend on enforcement by centralized authorities
- Can span jurisdictions, thus allowing for trade in ways that previously was expensive/difficult/illegal
- Cryptocurrencies enable this feat through technology, mathematics, and a philosophical innovation
- Property rights need not hold with certainty, it is sufficient that they hold probabilistically
- How the digital signature, blockchain, and PoW enabled property rights and bitcoin to operate

Closing Thoughts

- The Bitcoin system is a problem of collective memory
 - Relies on the truthfulness of collective memory to function well
 - Collective memory can be tampered with, given enough power, to disastrous effects



Wharton
UNIVERSITY of PENNSYLVANIA

ONLINE