The Economic Limits of Bitcoin and Anonymous, Decentralized Trust on the Blockchain

Eric Budish
Chicago Booth

June 2022

Nakamoto's Invention

- Satoshi Nakamoto invented a new kind of trust
- Completely anonymous and decentralized
- Without support from traditional sources: rule of law, reputations, relationships, collateral, trusted intermediaries

Nakamoto's Invention

- Satoshi Nakamoto invented a new kind of trust
- Completely anonymous and decentralized
- Without support from traditional sources: rule of law, reputations, relationships, collateral, trusted intermediaries
- At a high level: Nakamoto invented an elaborate scheme, combining ideas from CS+Econ, to incentivize a large mass of compute power around the world to pay attention to and collectively maintain a common data set
  - Enabling trust in this data set

Nakamoto's Invention

- Satoshi Nakamoto invented a new kind of trust
- Completely anonymous and decentralized
- Without support from traditional sources: rule of law, reputations, relationships, collateral, trusted intermediaries
- At a high level: Nakamoto invented an elaborate scheme, combining ideas from CS+Econ, to incentivize a large mass of compute power around the world to pay attention to and collectively maintain a common data set
  - Enabling trust in this data set
- This invention enabled cryptocurrencies — including Nakamoto's own Bitcoin
- The specific data structure maintained is called a blockchain

Nakamoto's Invention

- Nakamoto's invention captured the world's attention
- Recent peak: $3 trillion
- Even this figure seems to understate the amount of cultural, political and commercial attention that has been paid to blockchains and cryptocurrencies

Nakamoto's Invention

- Nakamoto's invention captured the world's attention
- Recent peak: $3 trillion
- Even this figure seems to understate the amount of cultural, political and commercial attention that has been paid to blockchains and cryptocurrencies
- Yet, usefulness remains an open question
- To date, majority of volume appears speculative, with other widely-documented use case being black market (Makarov and Schoar, Foley et al, speeches by Gensler, Yellen)
  - Ironically, most of the speculative volume is through cryptocurrency exchanges — which are trusted financial intermediaries

Nakamoto's Invention

- Nakamoto's invention captured the world's attention
- Recent peak: $3 trillion
- Even this figure seems to understate the amount of cultural, political and commercial attention that has been paid to blockchains and cryptocurrencies
- Yet, usefulness remains an open question
- To date, majority of volume appears speculative, with other widely-documented use case being black market (Makarov and Schoar, Foley et al, speeches by Gensler, Yellen)
  - Ironically, most of the speculative volume is through cryptocurrency exchanges — which are trusted financial intermediaries
- Meanwhile: Bitcoin mining 0.3-0.8% of global electricity consumption, $15bn of deadweight loss per year to secure its trust. Ethereum another $15bn per year to secure its trust.

- U.S. Treasury Secretary, Janet Yellen, in Feb. 2021:

  "I don't think that bitcoin ... is widely used as a transaction mechanism ...To the extent it is used I fear it's often for illicit finance. ... It is a highly speculative asset."

- U.S. Treasury Secretary, Janet Yellen, in Feb. 2021:

  "I don't think that bitcoin ... is widely used as a transaction mechanism ...To the extent it is used I fear it's often for illicit finance. ... It is a highly speculative asset."

- U.S. SEC Chair, Gary Gensler, in Aug. 2021:

- U.S. Treasury Secretary, Janet Yellen, in Feb. 2021:

  "I don't think that bitcoin ... is widely used as a transaction mechanism ...To the extent it is used I fear it's often for illicit finance. ... It is a highly speculative asset."

- U.S. SEC Chair, Gary Gensler, in Aug. 2021:

  "Primarily, crypto assets provide digital, scarce vehicles for speculative investment. ... These assets haven't been used much as a unit of account. We also haven't seen crypto used much as a medium of exchange. To the extent that it is used as such, it's often to skirt our laws with respect to anti-money laundering, sanctions, and tax collection. It also can enable extortion via ransomware ..."

This Paper's Argument

- This paper argues that Bitcoin and the Nakamoto blockchain — while undeniably ingenious — have serious economic limitations

This Paper's Argument

- This paper argues that Bitcoin and the Nakamoto blockchain — while undeniably ingenious — have serious economic limitations
- Cost of maintaining Nakamoto's anonymous, decentralized trust:

This Paper's Argument

- This paper argues that Bitcoin and the Nakamoto blockchain — while undeniably ingenious — have serious economic limitations
- Cost of maintaining Nakamoto's anonymous, decentralized trust:
  1. Very large in absolute terms relative to the stakes involved

This Paper's Argument

- This paper argues that Bitcoin and the Nakamoto blockchain — while undeniably ingenious — have serious economic limitations
- Cost of maintaining Nakamoto's anonymous, decentralized trust:
  1. Very large in absolute terms relative to the stakes involved
  2. *Scales linearly* with the stakes involved.

This Paper's Argument

- This paper argues that Bitcoin and the Nakamoto blockchain — while undeniably ingenious — have serious economic limitations
- Cost of maintaining Nakamoto's anonymous, decentralized trust:
  1. Very large in absolute terms relative to the stakes involved
  2. *Scales linearly* with the stakes involved.
- Thus, *if* Bitcoin is to become significantly more economically useful (e.g., large part of global finance), *then* the cost of maintaining Bitcoin must grow commensurately as well

This Paper's Argument

- This paper argues that Bitcoin and the Nakamoto blockchain — while undeniably ingenious — have serious economic limitations
- Cost of maintaining Nakamoto's anonymous, decentralized trust:
  1. Very large in absolute terms relative to the stakes involved
  2. *Scales linearly* with the stakes involved.
- Thus, *if* Bitcoin is to become significantly more economically useful (e.g., large part of global finance), *then* the cost of maintaining Bitcoin must grow commensurately as well
- Suggests skepticism that Bitcoin and its anonymous, decentralized trust will play a major role in the global economy and financial system

This Paper's Argument

- Core of argument is just 3 equations. Amount of computational work must simultaneously satisfy:

This Paper's Argument

- Core of argument is just 3 equations. Amount of computational work must simultaneously satisfy:
  - (1) zero-profits condition: for blockchain "miners"

This Paper's Argument

- Core of argument is just 3 equations. Amount of computational work must simultaneously satisfy:
  - (1) zero-profits condition: for blockchain "miners"
  - (2) incentive compatibility condition: deter "majority attack"

This Paper's Argument

- Core of argument is just 3 equations. Amount of computational work must simultaneously satisfy:
  - (1) zero-profits condition: for blockchain "miners"
  - (2) incentive compatibility condition: deter "majority attack"
- Together, (1)+(2) imply:

This Paper's Argument

- Core of argument is just 3 equations. Amount of computational work must simultaneously satisfy:
  - (1) zero-profits condition: for blockchain "miners"
  - (2) incentive compatibility condition: deter "majority attack"
- Together, (1)+(2) imply:
  - (3) recurring, "flow" payments to miners for maintaining the blockchain must be large relative to one-off, "stock" benefits of attacking the blockchain

This Paper's Argument

- Core of argument is just 3 equations. Amount of computational work must simultaneously satisfy:
  - (1) zero-profits condition: for blockchain "miners"
  - (2) incentive compatibility condition: deter "majority attack"
- Together, (1)+(2) imply:
  - (3) recurring, "flow" payments to miners for maintaining the blockchain must be large relative to one-off, "stock" benefits of attacking the blockchain
  - Very expensive!

This Paper's Argument

- Core of argument is just 3 equations. Amount of computational work must simultaneously satisfy:
  - (1) zero-profits condition: for blockchain "miners"
  - (2) incentive compatibility condition: deter "majority attack"
- Together, (1)+(2) imply:
  - (3) recurring, "flow" payments to miners for maintaining the blockchain must be large relative to one-off, "stock" benefits of attacking the blockchain
  - Very expensive!
  - Like a large implicit tax. Example calculations
    - To secure against a $1M attack: $3-$31 per transaction implicit tax

This Paper's Argument

- Core of argument is just 3 equations. Amount of computational work must simultaneously satisfy:
  - (1) zero-profits condition: for blockchain "miners"
  - (2) incentive compatibility condition: deter "majority attack"
- Together, (1)+(2) imply:
  - (3) recurring, "flow" payments to miners for maintaining the blockchain must be large relative to one-off, "stock" benefits of attacking the blockchain
  - Very expensive!
  - Like a large implicit tax. Example calculations
    - To secure against a $1M attack: $3-$31 per transaction implicit tax
    - To secure against a $1B attack: $3k-31k per transaction!

This Paper's Argument

- Core of argument is just 3 equations. Amount of computational work must simultaneously satisfy:
  - (1) zero-profits condition: for blockchain "miners"
  - (2) incentive compatibility condition: deter "majority attack"
- Together, (1)+(2) imply:
  - (3) recurring, "flow" payments to miners for maintaining the blockchain must be large relative to one-off, "stock" benefits of attacking the blockchain
  - Very expensive!
  - Like a large implicit tax. Example calculations
    - To secure against a $1M attack: $3-$31 per transaction implicit tax
    - To secure against a $1B attack: $3k-31k per transaction!
- A way out of this argument:

This Paper's Argument

- Core of argument is just 3 equations. Amount of computational work must simultaneously satisfy:
  - (1) zero-profits condition: for blockchain "miners"
  - (2) incentive compatibility condition: deter "majority attack"
- Together, (1)+(2) imply:
  - (3) recurring, "flow" payments to miners for maintaining the blockchain must be large relative to one-off, "stock" benefits of attacking the blockchain
  - Very expensive!
  - Like a large implicit tax. Example calculations
    - To secure against a $1M attack: $3-$31 per transaction implicit tax
    - To secure against a $1B attack: $3k-31k per transaction!
- A way out of this argument:
  - (i) mining technology is specialized/non-repurposable, and
  - (ii) any majority attack is a "sabotage", causes collapse

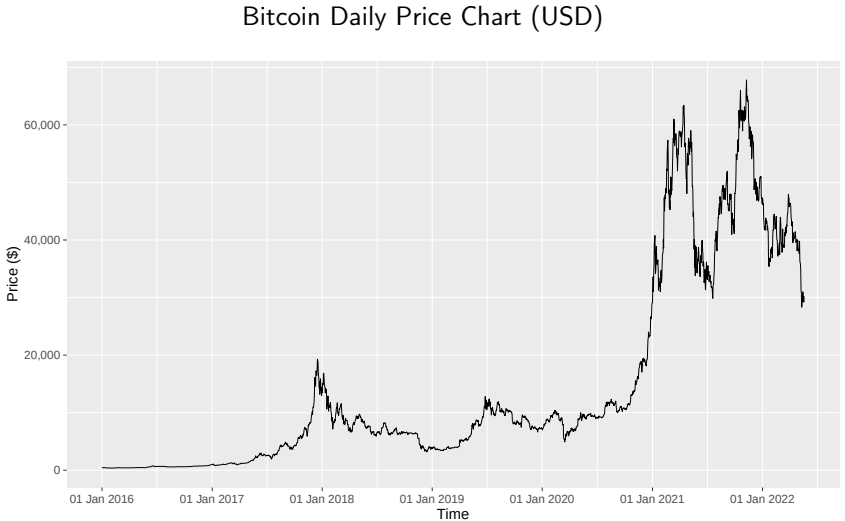
This Paper's Argument

- Core of argument is just 3 equations. Amount of computational work must simultaneously satisfy:
  - (1) zero-profits condition: for blockchain "miners"
  - (2) incentive compatibility condition: deter "majority attack"
- Together, (1)+(2) imply:
  - (3) recurring, "flow" payments to miners for maintaining the blockchain must be large relative to one-off, "stock" benefits of attacking the blockchain
  - Very expensive!
  - Like a large implicit tax. Example calculations
    - To secure against a $1M attack: $3-$31 per transaction implicit tax
    - To secure against a $1B attack: $3k-31k per transaction!
- A way out of this argument:
  - (i) mining technology is specialized/non-repurposable, and
  - (ii) any majority attack is a "sabotage", causes collapse
- But: vulnerability to sabotage is itself a serious concern. "Pick your poison"

This Paper's Argument

- Core of argument is just 3 equations. Amount of computational work must simultaneously satisfy:
  - (1) zero-profits condition: for blockchain "miners"
  - (2) incentive compatibility condition: deter "majority attack"
- Together, (1)+(2) imply:
  - (3) recurring, "flow" payments to miners for maintaining the blockchain must be large relative to one-off, "stock" benefits of attacking the blockchain
  - Very expensive!
  - Like a large implicit tax. Example calculations
    - To secure against a $1M attack: $3-$31 per transaction implicit tax
    - To secure against a $1B attack: $3k-31k per transaction!
- A way out of this argument:
  - (i) mining technology is specialized/non-repurposable, and
  - (ii) any majority attack is a "sabotage", causes collapse
- But: vulnerability to sabotage is itself a serious concern. "Pick your poison"
- Analysis points to specific collapse scenarios

A Disclaimer ...



Source: www.wsj.com

A Disclaimer ...

- I have no explanation for Bitcoin's current asset value (currrently about $600bn, peak of >$1tn).
- My paper shows that Bitcoin's economic usefulness is likely to continue to be limited.
- A natural conjecture is low economic usefulness implies low asset value (Athey et al., 2016), but these are separable things
- Also confusing: a large majority of volume at present is on crypto exchanges
  - Not anonymous, decentralized trust as invented by Nakamoto.
  - Rather: trusting Coinbase + Rule-of-law instead of JPMorgan + Rule-of-law.
  - Seems mostly speculation (Makarov and Schoar, 2022)
  - I don't get it, but that is separate from my paper.
- To date: the claim that Bitcoin's economic usefulness is limited is still looking good, despite the high market capitalization. But we'll see.

Overview of the Talk

- Overview: What is Nakamoto Blockchain
- Nakamoto Blockchain: A Critique in 3 Equations
  - Rent-Seeking Competition (Miners)
  - Incentive Compatibility (Majority Attack)
  - Economic Constraint on the Blockchain: Flow vs. Stock
- Majority Attack Scenarios
  - Double Spending
  - Sabotage
- Blockchain-Specific Mining Technology
  - A Softer Constraint: Stock vs. Stock
  - Collapse Scenarios

What is Nakamoto Blockchain (1/4)

- **Transaction:** sender, receiver, amount, signature

Sender	Receiver	Amount	Signature
Alice	Bob	$10	*Alice*

- **Signature**:
  - Proves sender's identity
  - Encodes transaction details (amount, recipient)
  - Standard cryptography techniques

What is Nakamoto Blockchain (1/4)

- **Transaction:** sender, receiver, amount, signature

Sender	Receiver	Amount	Signature
Alice	Bob	$10	*Alice*

- **Signature**:
  - Proves sender's identity
  - Encodes transaction details (amount, recipient)
  - Standard cryptography techniques
- Imagine transactions on a google spreadsheet
  - Signature: only Alice can add transactions in which Alice sends money
  - But:
    - Alice can send money she doesn't have
    - Alice can send money she does have but to multiple parties at the same time
    - Alice can delete previous transactions (her own or others')

What is Nakamoto Blockchain (1/4)

- **Transaction:** sender, receiver, amount, signature

Sender	Receiver	Amount	Signature
Alice	Bob	$10	*Alice*

- **Signature**:
  - Proves sender's identity
  - Encodes transaction details (amount, recipient)
  - Standard cryptography techniques
- Imagine transactions on a google spreadsheet
  - Signature: only Alice can add transactions in which Alice sends money
  - But:
    - Alice can send money she doesn't have
    - Alice can send money she does have but to multiple parties at the same time
    - Alice can delete previous transactions (her own or others')
- Imagine transactions through a trusted party that keeps track of balances
  - That works just fine re: security issues listed above
  - But: requires a trusted party.

What is Nakamoto Blockchain (2/4)

Nakamoto (2008) Blockchain Innovation

What is Nakamoto Blockchain (2/4)

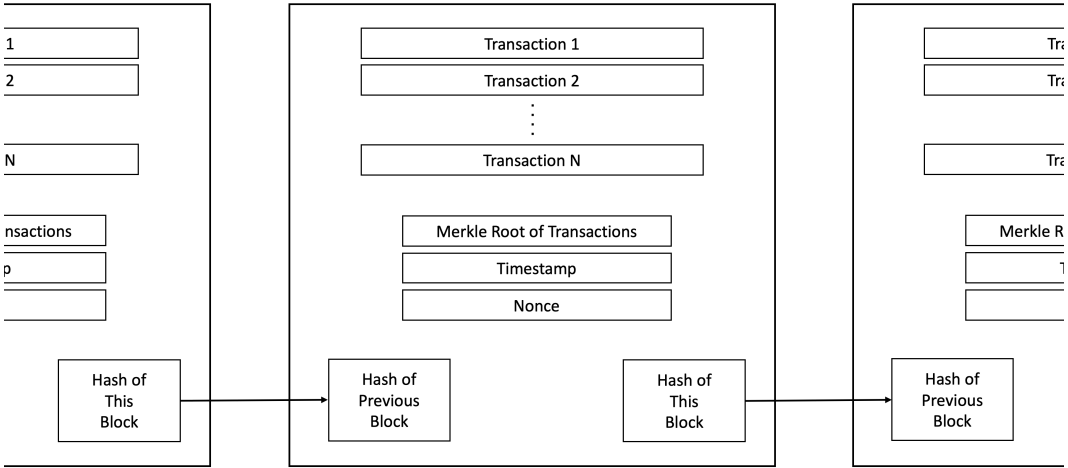
Nakamoto (2008) Blockchain Innovation

- **I: Pending Transactions List**
  - Users submit transactions to a pending transactions list, called mempool
  - Like a google spreadsheet — not considered official yet

What is Nakamoto Blockchain (2/4)

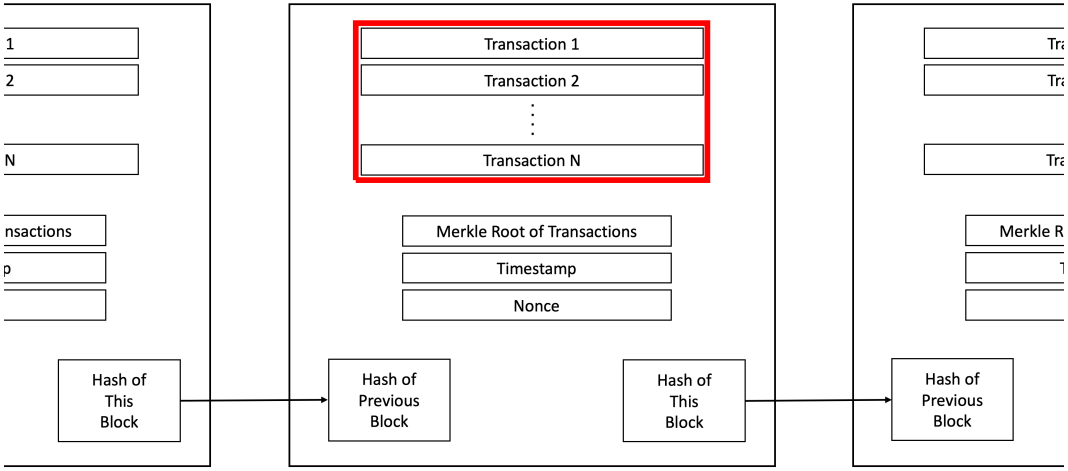
Nakamoto (2008) Blockchain Innovation

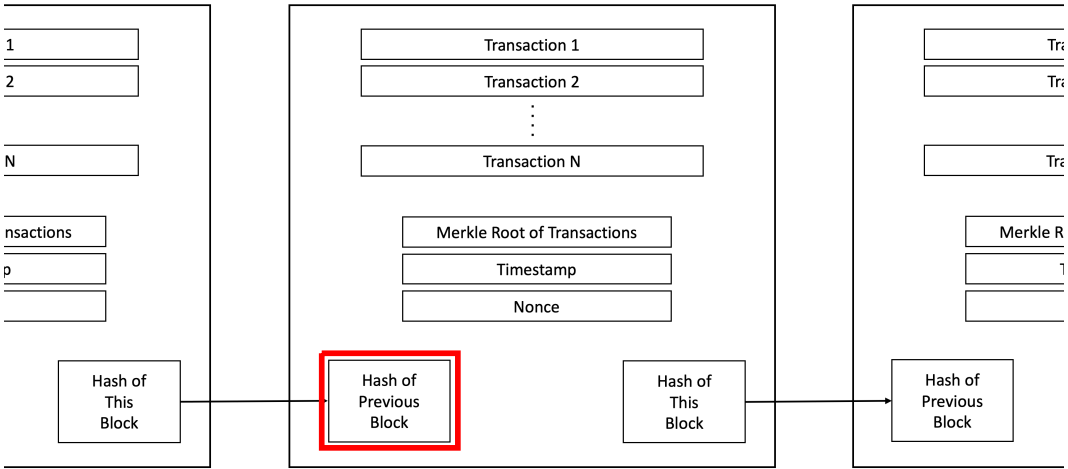
- **I: Pending Transactions List**
  - Users submit transactions to a pending transactions list, called mempool
  - Like a google spreadsheet — not considered official yet
- **II: Valid Blocks**
  - Bitcoin "miners" compete for the right to add transactions from the mempool to a data structure called the blockchain. (Will describe competition next)
  - Blocks consist of $\approx$ 2000 transactions
  - Each new block "chains" to previous block, by including a hash of the data in the previous block (Haber and Stornetta, 1991)
  - Validity: for a block to be valid:
    1. Each individual transaction must be properly signed
    2. Each individual trandaction must be funded given previous blocks
    3. No contradictions: there cannot be multiple transactions sending the same funds


1
2
N
nsactions
p
Transaction 1
Transaction 2
Transaction N
Merkle Root of Transactions
Timestamp
Nonce
Tra
Tra
Tra
Merkle R
Hash of This Block
Hash of Previous Block
Hash of This Block
Hash of Previous Block

Conditions for a Valid Block:

1. **Each individual transaction correctly signed,**
2. **Each individual transaction funded given history,**
3. **No contradictions in the set of transactions.**





Any change to history changes the hash of the previous block.

What is Nakamoto Blockchain (3/4)

- **III: Bitcoin Mining Computational Tournament**
  - Boils down to a massive brute-force search for a lucky random alphanumeric string
  - Bitcoin miners choose a valid block of transactions from the mempool that they wish to chain to the previous block
  - Then search for an alphanumeric string ("nonce"), such that, when all of the data is hashed together using SHA-256, the result has a large number of leading zeros
  - Called "proof of work" – hard to find, easy to check
  - Current hash rate: about 200 million TH/s (2 x 10^20)
  - Example: block 729,999 has the hash

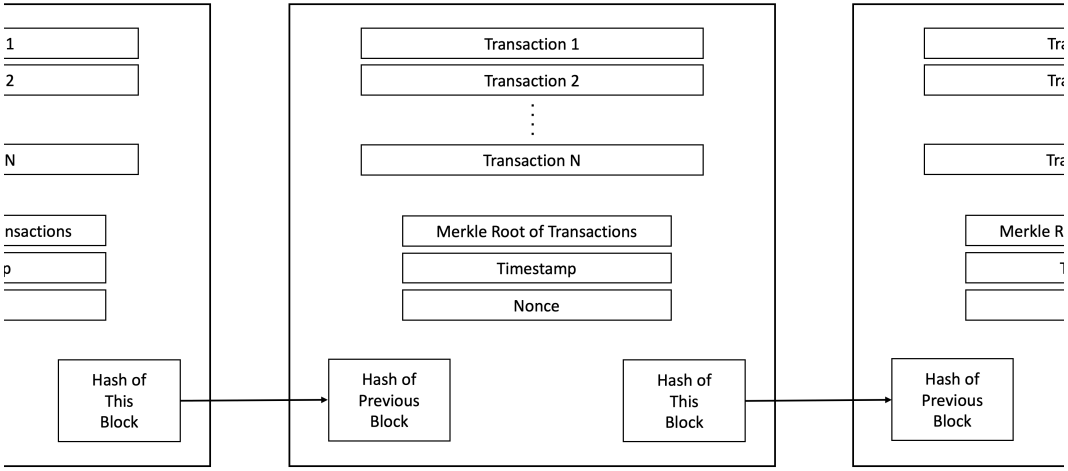
$$0000000000000000000008b6f6fb83f8d74512ef1e0af29e642dd20daddd7d318f$$

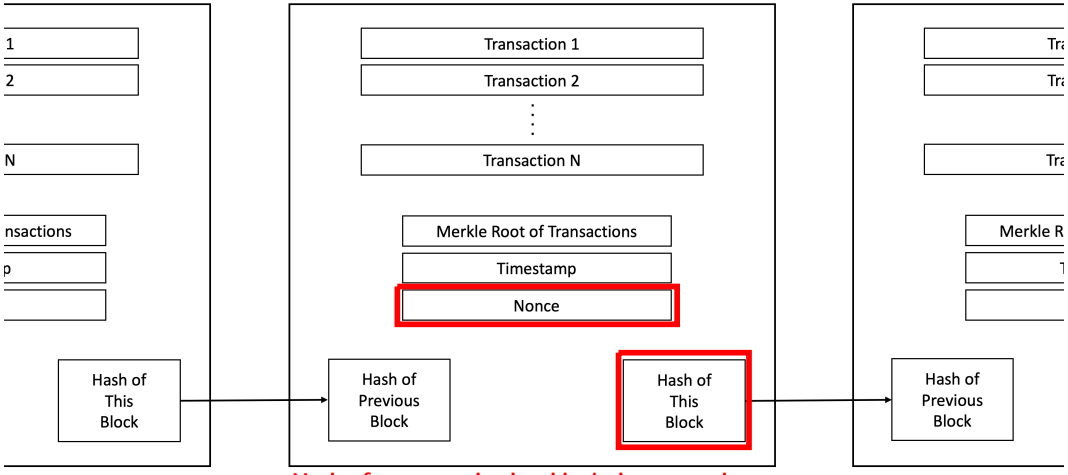
What is Nakamoto Blockchain (3/4)

- **III: Bitcoin Mining Computational Tournament**
  - Boils down to a massive brute-force search for a lucky random alphanumeric string
  - Bitcoin miners choose a valid block of transactions from the mempool that they wish to chain to the previous block
  - Then search for an alphanumeric string ("nonce"), such that, when all of the data is hashed together using SHA-256, the result has a large number of leading zeros
  - Called "proof of work" – hard to find, easy to check
  - Current hash rate: about 200 million TH/s (2 x 10^20)
  - Example: block 729,999 has the hash

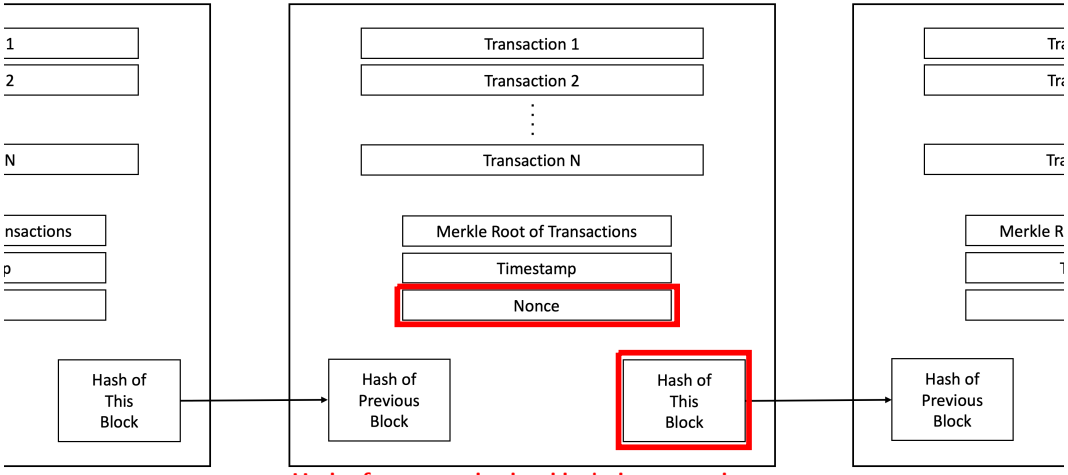
$$0000000000000000008b6f6fb83f8d74512ef1e0af29e642dd20daddd7d318f$$

- Miner who finds a lucky hash broadcasts their new block
  - Other miners check validity (fast), then start working on the next block
  - Winner earns reward paid in bitcoin ("block reward" $\approx$ \$250k)
- Tournament difficulty calibrated to take about 10 minutes


1
2
N
nsactions
p
Hash of This Block
Transaction 1
Transaction 2
Transaction N
Merkle Root of Transactions
Timestamp
Nonce
Hash of Previous Block
Hash of This Block
Tra
Tra
Tra
Merkle R
Hash of Previous Block



Hash of nonce and other block data must have a very large number of leading zeros.



Hash of nonce and other block data must have a very large number of leading zeros.
Example from Block $729,999$**:**
- **Nonce:** 3477019455
- **Hash:** $000000000000000000008b6f6fb83f8d745\ldots$

What is Nakamoto Blockchain (4/4)

- **IV Longest-Chain Convention**
  - Once a miner finds a lucky alphanumeric string, all miners are supposed to move on to mining the next block
  - To induce this, Nakamoto proposed the longest-chain convention
    - If there are multiple chains, then the longest chain, as measured by the amount of computational work, is the official consensus record of transactions

What is Nakamoto Blockchain (4/4)

- **IV Longest-Chain Convention**
  - Once a miner finds a lucky alphanumeric string, all miners are supposed to move on to mining the next block
  - To induce this, Nakamoto proposed the longest-chain convention
    - If there are multiple chains, then the longest chain, as measured by the amount of computational work, is the official consensus record of transactions
  - Intuitively: provides incentive for miners to focus their efforts on the current longest chain
    - (Note, multiple chains will happen by coincidence with some frequency: if 1 second latency, roughly 1/600 blocks)
  - Game-theoretic validity has received signficant attention:
    - Kroll et al (2013), Carlsten et al (2016), Biais et al (2019)
    - All of these works explicitly assume that miners are "small" — that is, they assume away majority attack

What is Nakamoto Blockchain: Summary

- From the Nakamoto (2008) abstract:

What is Nakamoto Blockchain: Summary

- From the Nakamoto (2008) abstract:

  "We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work.

What is Nakamoto Blockchain: Summary

- From the Nakamoto (2008) abstract:

  "We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain serves not only as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power.

What is Nakamoto Blockchain: Summary

- From the Nakamoto (2008) abstract:

  "We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain serves not only as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers." (Emphasis added)

What is Nakamoto Blockchain: Summary

- From the Nakamoto (2008) abstract:

  "We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain serves not only as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers." (Emphasis added)

- The abstract succinctly summarizes the accomplishment and its vulnerability

What is Nakamoto Blockchain: Summary

- From the Nakamoto (2008) abstract:

  "We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain serves not only as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers." (Emphasis added)

- The abstract succinctly summarizes the accomplishment and its vulnerability
- Anonymous, decentralized trust. A "purely peer-to-peer version of electronic cash" without "a trusted third party ... to prevent double-spending"

What is Nakamoto Blockchain: Summary

- From the Nakamoto (2008) abstract:

  "We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain serves not only as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers." (Emphasis added)

- The abstract succinctly summarizes the accomplishment and its vulnerability
- Anonymous, decentralized trust. A "purely peer-to-peer version of electronic cash" without "a trusted third party ... to prevent double-spending"
- But, vulnerable to majority attack.

Clarification

- As interest in Bitcoin and its blockchain have surged, some have started to use the phrase "blockchain" to describe distributed databases among *known, trusted parties* – that is, *without* the central innovation of Nakamoto (2008)

Clarification

- As interest in Bitcoin and its blockchain have surged, some have started to use the phrase "blockchain" to describe distributed databases among *known, trusted parties* – that is, *without* the central innovation of Nakamoto (2008)

  "If you announce that you are updating the database software used by a consortium of banks to track derivatives trades, the New York Times will not write an article about it. If you say that you are blockchaining the blockchain software used by a blockchain of blockchains to blockchain blockchain blockchains, the New York Times will blockchain a blockchain about it." (Matt Levine, 2017)

Clarification

- As interest in Bitcoin and its blockchain have surged, some have started to use the phrase "blockchain" to describe distributed databases among *known, trusted parties* – that is, *without* the central innovation of Nakamoto (2008)

  "If you announce that you are updating the database software used by a consortium of banks to track derivatives trades, the New York Times will not write an article about it. If you say that you are blockchaining the blockchain software used by a blockchain of blockchains to blockchain blockchain blockchains, the New York Times will blockchain a blockchain about it." (Matt Levine, 2017)

- My critique is of blockchain in the sense of Nakamoto (2008), not of distributed databases / ledgers
- A very interesting open question is what exists "in between" these two poles. Will return to this at the end time permitting

Overview of the Talk

- Overview: What is Nakamoto Blockchain
- **Nakamoto Blockchain: A Critique in 3 Equations**
  - **Rent-Seeking Competition (Miners)**
  - **Incentive Compatibility (Majority Attack)**
  - **Economic Constraint on the Blockchain: Flow vs. Stock**
- Majority Attack Scenarios
  - Double Spending
  - Sabotage
- Blockchain-Specific Mining Technology
  - A Softer Constraint: Stock vs. Stock
  - Collapse Scenarios

Rent-Seeking Competition (Miners)

- p_{block} : economic reward to miner who wins computational tournament
  - Assume exogenous; will place constraints below

Rent-Seeking Competition (Miners)

- p_{block} : economic reward to miner who wins computational tournament
  - Assume exogenous; will place constraints below
- c: cost per unit time of one unit of computational power
  - Per-block electricity costs + Per-block cost of capital, incl. depreciation.
  - Unit of time = amount it takes to mine one block if everyone honest (10 minutes for Bitcoin)

Rent-Seeking Competition (Miners)

- p_{block} : economic reward to miner who wins computational tournament
  - Assume exogenous; will place constraints below
- c: cost per unit time of one unit of computational power
  - Per-block electricity costs + Per-block cost of capital, incl. depreciation.
  - Unit of time = amount it takes to mine one block if everyone honest (10 minutes for Bitcoin)
- Note: rental cost of capital is appropriate for honest mining whether the capital is repurposable or specialized
  - Repurposable: initial Nakamoto ideal of "one-CPU-one-vote"
  - Specialized: ASICs, predominant since mid-2013. (Prat and Walter, 2021)

Rent-Seeking Competition (Miners)

- p_{block} : economic reward to miner who wins computational tournament
  - Assume exogenous; will place constraints below
- c: cost per unit time of one unit of computational power
  - Per-block electricity costs + Per-block cost of capital, incl. depreciation.
  - Unit of time = amount it takes to mine one block if everyone honest (10 minutes for Bitcoin)
- Note: rental cost of capital is appropriate for honest mining whether the capital is repurposable or specialized
  - Repurposable: initial Nakamoto ideal of "one-CPU-one-vote"
  - Specialized: ASICs, predominant since mid-2013. (Prat and Walter, 2021)
- N units of computational power $\rightarrow \frac{1}{N}$ prob of winning p_{block}

Rent-Seeking Competition (Miners)

- p_{block} : economic reward to miner who wins computational tournament
  - Assume exogenous; will place constraints below
- c: cost per unit time of one unit of computational power
  - Per-block electricity costs + Per-block cost of capital, incl. depreciation.
  - Unit of time = amount it takes to mine one block if everyone honest (10 minutes for Bitcoin)
- Note: rental cost of capital is appropriate for honest mining whether the capital is repurposable or specialized
  - Repurposable: initial Nakamoto ideal of "one-CPU-one-vote"
  - Specialized: ASICs, predominant since mid-2013. (Prat and Walter, 2021)
- N units of computational power $\rightarrow \frac{1}{N}$ prob of winning p_{block}
- Honest mining, Free entry equilibrium

$$N^* c = p_{block} \tag{1}$$

- Note: (1) widely known (many papers, Bitcoin Wiki)

Incentive Compatibility (Majority Attack)

- Well-known that blockchain vulnerable to majority attack
- Abstract of Nakamoto (2008):

  "The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain serves not only as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers." (Emphasis added)

- Bitcoin Wiki:

  "Bitcoin's security model relies on no single coalition of miners controlling more than half the mining power"

Incentive Compatibility (Majority Attack)

- What is cost of a majority?

Incentive Compatibility (Majority Attack)

- What is cost of a majority?
- Insider: $\frac{N^*}{2} + \epsilon$ units of compute power, at cost $(\frac{N^*}{2} + \epsilon)c$ per unit time
- Outsider: $(N^* + \epsilon)c$ per unit time
  - Analysis will mostly focus on costs of outsider attacks (conservative, conceptually nicer)
  - Practically, insider attacks may be more worrisome; also cheaper

Incentive Compatibility (Majority Attack)

- What is cost of a majority?
- Insider: $\frac{N^*}{2} + \epsilon$ units of compute power, at cost $(\frac{N^*}{2} + \epsilon)c$ per unit time
- Outsider: $(N^* + \epsilon)c$ per unit time
  - Analysis will mostly focus on costs of outsider attacks (conservative, conceptually nicer)
  - Practically, insider attacks may be more worrisome; also cheaper
- Suppose it takes outside attacker with $\frac{A}{A+1}$ majority t time in expectation
  - Will depend on escrow period, nature of attack (computed below for double spending)
- Let $At \cdot N^*c$ denote the gross cost of attack

Incentive Compatibility (Majority Attack)

- What is cost of a majority?
- Insider: $\frac{N^*}{2} + \epsilon$ units of compute power, at cost $(\frac{N^*}{2} + \epsilon)c$ per unit time
- Outsider: $(N^* + \epsilon)c$ per unit time
  - Analysis will mostly focus on costs of outsider attacks (conservative, conceptually nicer)
  - Practically, insider attacks may be more worrisome; also cheaper
- Suppose it takes outside attacker with $\frac{A}{A+1}$ majority t time in expectation
  - Will depend on escrow period, nature of attack (computed below for double spending)
- Let $At \cdot N^*c$ denote the gross cost of attack
- Let V_{attack} denote the value of an attack
  - For now, abstract
  - Other than value will grow as Bitcoin's usefulness and importance grow
- Initial (gross) version of incentive constraint:

$$At \cdot N^*c > V_{attack}$$

Incentive Compatibility (Majority Attack)

- What I will call net cost of attack differs from gross costs for three reasons

Incentive Compatibility (Majority Attack)

- What I will call net cost of attack differs from gross costs for three reasons
- Reason 1: Attacker earns block rewards from the attack
  - An A attacker who mines for t time earns At block rewards in expectation
  - Worth $At \cdot p_{block} = At \cdot N^* c$ per equation (1)

Incentive Compatibility (Majority Attack)

- What I will call net cost of attack differs from gross costs for three reasons
- Reason 1: Attacker earns block rewards from the attack
  - An A attacker who mines for t time earns At block rewards in expectation
  - Worth $At \cdot p_{block} = At \cdot N^* c$ per equation (1)
- Reason 2: Attacker may face frictions relative to honest miners
  - Ex: attacker compute power may be less energy efficient, start/stop costs
  - Let $\kappa \geq 0$ parameterize cost inefficiency, s.t. cost is $(1 + \kappa)At \cdot N^* c$

Incentive Compatibility (Majority Attack)

- What I will call net cost of attack differs from gross costs for three reasons
- Reason 1: Attacker earns block rewards from the attack
  - An A attacker who mines for t time earns At block rewards in expectation
  - Worth $At \cdot p_{block} = At \cdot N^* c$ per equation (1)
- Reason 2: Attacker may face frictions relative to honest miners
  - Ex: attacker compute power may be less energy efficient, start/stop costs
  - Let $\kappa \geq 0$ parameterize cost inefficiency, s.t. cost is $(1 + \kappa)At \cdot N^* c$
- Reason 3: Attack may harm post-attack value of Bitcoin
  - This increases attacker's costs (Bitcoin holdings, specialized capital)
  - But opens up possibility of attack motivated by this harm per se
  - Will discuss in detail under Sabotage. Here, assume no harm.

Incentive Compatibility (Majority Attack)

- What I will call net cost of attack differs from gross costs for three reasons
- Reason 1: Attacker earns block rewards from the attack
  - An A attacker who mines for t time earns At block rewards in expectation
  - Worth $At \cdot p_{block} = At \cdot N^* c$ per equation (1)
- Reason 2: Attacker may face frictions relative to honest miners
  - Ex: attacker compute power may be less energy efficient, start/stop costs
  - Let $\kappa \geq 0$ parameterize cost inefficiency, s.t. cost is $(1+\kappa)At \cdot N^* c$
- Reason 3: Attack may harm post-attack value of Bitcoin
  - This increases attacker's costs (Bitcoin holdings, specialized capital)
  - But opens up possibility of attack motivated by this harm per se
  - Will discuss in detail under Sabotage. Here, assume no harm.
- Net cost is $(1+\kappa)At \cdot N^* c - At \cdot p_{block} = \kappa At \cdot N^* c$. Net IC constraint:

$$\kappa At \cdot N^* c > V_{attack} \tag{2}$$

Incentive Compatibility (Majority Attack)

- What I will call net cost of attack differs from gross costs for three reasons
- Reason 1: Attacker earns block rewards from the attack
  - An A attacker who mines for t time earns At block rewards in expectation
  - Worth $At \cdot p_{block} = At \cdot N^* c$ per equation (1)
- Reason 2: Attacker may face frictions relative to honest miners
  - Ex: attacker compute power may be less energy efficient, start/stop costs
  - Let $\kappa \geq 0$ parameterize cost inefficiency, s.t. cost is $(1+\kappa)At \cdot N^* c$
- Reason 3: Attack may harm post-attack value of Bitcoin
  - This increases attacker's costs (Bitcoin holdings, specialized capital)
  - But opens up possibility of attack motivated by this harm per se
  - Will discuss in detail under Sabotage. Here, assume no harm.
- Net cost is $(1+\kappa)At \cdot N^* c - At \cdot p_{block} = \kappa At \cdot N^* c$. Net IC constraint:

$$\kappa At \cdot N^* c > V_{attack} \tag{2}$$

- Proposition: if the attacker's cost is the same as honest miners ($\kappa = 0$) and the attack does not cause the value of Bitcoin to fall, then net cost of attack is zero.

Critique in 3 Equations

The Problem

$$N^* c = p_{block} \tag{1}$$

$$\kappa A t \cdot N^* c > V_{attack} \tag{2}$$

- Together (1) and (2) imply:

Critique in 3 Equations

The Problem

$$N^* c = p_{block} \tag{1}$$

$$\kappa At \cdot N^* c > V_{attack} \tag{2}$$

- Together (1) and (2) imply:

$$p_{block} > \frac{V_{attack}}{\kappa At} \tag{3}$$

Critique in 3 Equations

The Problem

$$N^* c = p_{block} \tag{1}$$

$$\kappa A t \cdot N^* c > V_{attack} \tag{2}$$

- Together (1) and (2) imply:

$$p_{block} > \frac{V_{attack}}{\kappa A t} \tag{3}$$

- In words: *the equilibrium per-block payment to miners for maintaining the blockchain has to be large relative to the one-off benefits of attacking it*

Critique in 3 Equations

The Problem

$$N^* c = p_{block} \tag{1}$$

$$\kappa A t \cdot N^* c > V_{attack} \tag{2}$$

- Together (1) and (2) imply:

$$p_{block} > \frac{V_{attack}}{\kappa A t} \tag{3}$$

- In words: *the equilibrium per-block payment to miners for maintaining the blockchain has to be large relative to the one-off benefits of attacking it*
- Flow payment to miners > Stock value of attack

Critique in 3 Equations

$$p_{block} > \frac{V_{attack}}{\kappa A t}$$

- Remarks:

Critique in 3 Equations

$$p_{block} > \frac{V_{attack}}{\kappa A t}$$

- Remarks:
- Economics: *very expensive* form of trust. Memoryless.
  - Usual alternatives: reputations, relationships, collateral, rule-of-law.
  - Imagine a brand only secure as flow investment in advertising. Or a military only as secure as # of soldiers on border.

Critique in 3 Equations

$$p_{block} > \frac{V_{attack}}{\kappa A t}$$

- Remarks:
- Economics: *very expensive* form of trust. Memoryless.
  - Usual alternatives: reputations, relationships, collateral, rule-of-law.
  - Imagine a brand only secure as flow investment in advertising. Or a military only as secure as # of soldiers on border.
- Security: security is *linear* in amount of cpu power.
  - Example: a $1B attack is 1000x more expensive to prevent than a $1M attack.
  - Usual alternatives: cryptography, force, laws.
  - Imagine a company only as secure as the $ value of its cpu power.

Critique in 3 Equations

$$p_{block} > \frac{V_{attack}}{\kappa A t}$$

- Remarks:
- Equation (3) places potentially serious economic constraints on applicability of the Nakamoto (2008) blockchain:
  - The blockchain can only be used in economic contexts where users are willing to pay a per-block transactions cost, p_{block}, that is large relative to the value of a one-off attack, V_{attack}.
- By analogy, imagine if users of the Visa network had to pay fees to Visa, every ten minutes, that were large relative to the value of a successful one-off attack on the Visa network.

Overview of the Talk

- Overview: What is Nakamoto Blockchain
- Nakamoto Blockchain: A Critique in 3 Equations
  - Rent-Seeking Competition (Miners)
  - Incentive Compatibility (Majority Attack)
  - Economic Constraint on the Blockchain: Flow vs. Stock
- **Majority Attack Scenarios**
  - **Double Spending**
  - **Sabotage**
- Blockchain-Specific Mining Technology
  - A Softer Constraint: Stock vs. Stock
  - Collapse Scenarios

What Can An Attacker Do?

- A majority attacker can

What Can An Attacker Do?

- A majority attacker can
  - Solve computational puzzles faster, in expectation, than the honest minority

What Can An Attacker Do?

- A majority attacker can
  - Solve computational puzzles faster, in expectation, than the honest minority
  - Create an alternative longest chain, replace the honest chain at a strategically opportune moment

What Can An Attacker Do?

- A majority attacker can
  - Solve computational puzzles faster, in expectation, than the honest minority
  - Create an alternative longest chain, replace the honest chain at a strategically opportune moment
  - This allows the attacker to:

What Can An Attacker Do?

- A majority attacker can
  - Solve computational puzzles faster, in expectation, than the honest minority
  - Create an alternative longest chain, replace the honest chain at a strategically opportune moment
  - This allows the attacker to:
    - Control what transactions get added to the blockchain

What Can An Attacker Do?

- A majority attacker can
  - Solve computational puzzles faster, in expectation, than the honest minority
  - Create an alternative longest chain, replace the honest chain at a strategically opportune moment
  - This allows the attacker to:
    - Control what transactions get added to the blockchain
    - Remove recent transactions from the blockchain

What Can An Attacker Do?

- A majority attacker can
  - Solve computational puzzles faster, in expectation, than the honest minority
  - Create an alternative longest chain, replace the honest chain at a strategically opportune moment
  - This allows the attacker to:
    - Control what transactions get added to the blockchain
    - Remove recent transactions from the blockchain
  - The attacker also earns the block rewards, for each period of his alternative chain

What Can An Attacker Do?

- A majority attacker can
  - Solve computational puzzles faster, in expectation, than the honest minority
  - Create an alternative longest chain, replace the honest chain at a strategically opportune moment
  - This allows the attacker to:
    - Control what transactions get added to the blockchain
    - Remove recent transactions from the blockchain
  - The attacker also earns the block rewards, for each period of his alternative chain
- A majority attacker cannot
  - Create new transactions that spend other participants' Bitcoins ("steal all the Bitcoins")
  - This would require not just >50% majority, but breaking modern cryptography

Attack I: Double Spending

- Attacker can double spend:
  - (i) spend Bitcoins — i.e., engage in a transaction in which he sends Bitcoins to a merchant in exchange for goods or assets
  - (ii) allow that transaction to be added to the blockchain
  - (iii) the attacker works in secret to create an alternative longest chain (in which those same Bitcoins are sent to other accounts they control)
  - (iv) the attacker waits for any escrow periods to elapse, so they receive the goods or assets in (i)
  - (v) the attacker then releases their alternative longest chain. They now have the goods or assets received in (iv), and also the Bitcoins they sent to themselves in (iii)

Illustration of Double Spending

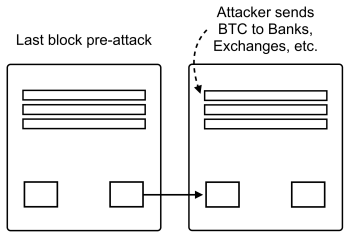


Illustration of Double Spending

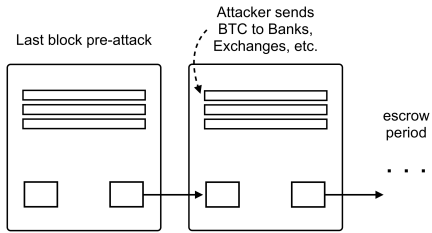


Illustration of Double Spending

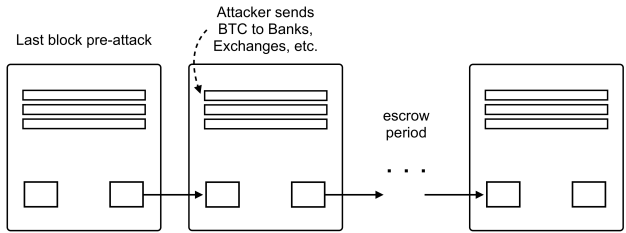


Illustration of Double Spending

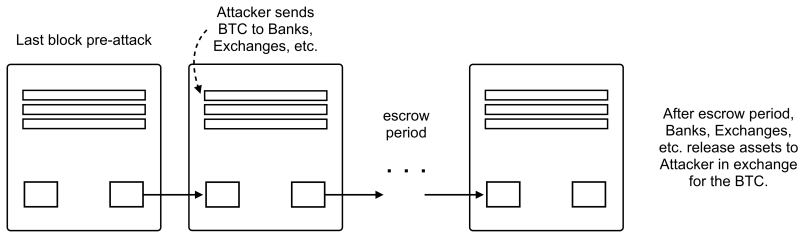


Illustration of Double Spending

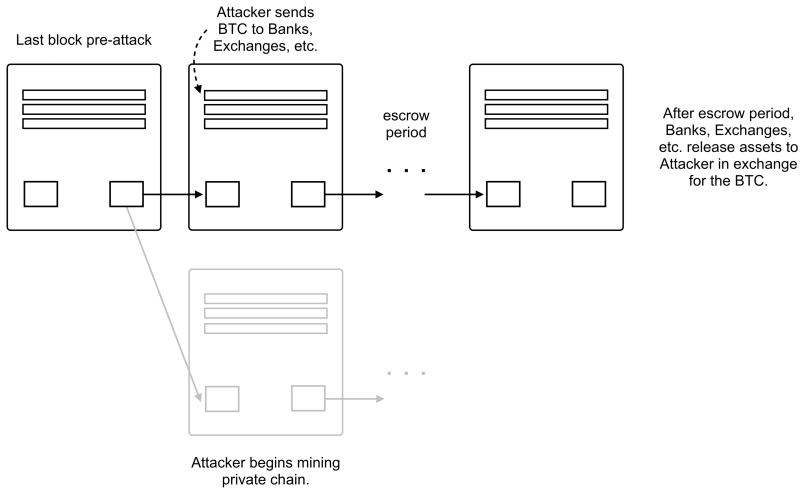


Illustration of Double Spending

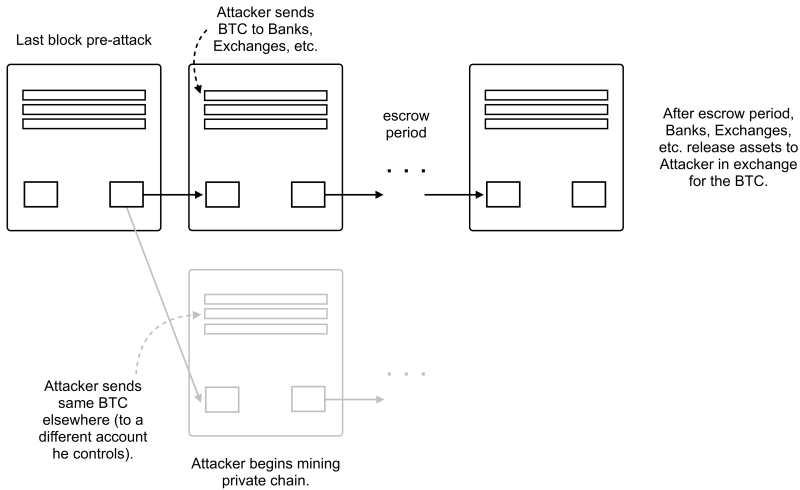


Illustration of Double Spending

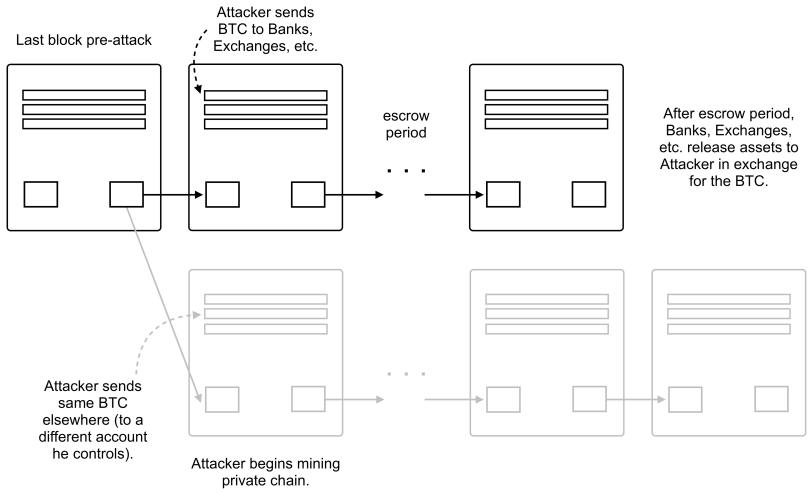
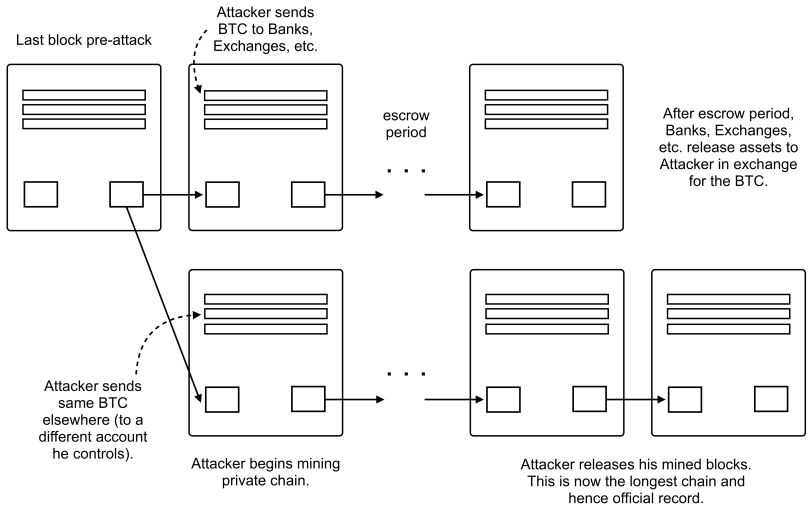


Illustration of Double Spending



Double Spending: Analysis Framework

- Equation (3) tells us that the possibility of a double-spending attack places economic constraints on Nakamoto trust
- To understand these constraints, we need to analyze
  - Benefits: V_{attack}
  - Costs: κAtN^*c

Double Spending: Analysis Framework

- Equation (3) tells us that the possibility of a double-spending attack places economic constraints on Nakamoto trust
- To understand these constraints, we need to analyze
  - Benefits: V_{attack}
  - Costs: $\kappa A t N^* c$
- Benefits
  - A majority attacker will not double-spend for a cappucino at Starbucks
  - They will use their majority to conduct transactions that are as large as possible given current uses of Nakamoto blockchain (potentially, many such transactions using many addresses)
  - Therefore, V_{attack} is a statistic on the amount of transaction volume that a large *honest* user of Bitcoin can conduct
  - I consider a range from \$1000 (pizza) to \$100bn (global finance)

Double Spending: Analysis Framework

- Equation (3) tells us that the possibility of a double-spending attack places economic constraints on Nakamoto trust
- To understand these constraints, we need to analyze
  - Benefits: V_{attack}
  - Costs: $\kappa A t N^* c$
- Benefits
  - A majority attacker will not double-spend for a cappucino at Starbucks
  - They will use their majority to conduct transactions that are as large as possible given current uses of Nakamoto blockchain (potentially, many such transactions using many addresses)
  - Therefore, V_{attack} is a statistic on the amount of transaction volume that a large *honest* user of Bitcoin can conduct
  - I consider a range from \$1000 (pizza) to \$100bn (global finance)
- Costs:
  - start by understanding t:

Double Spending: Attack Duration in Closed Form

- Let $t(A, e)$ denote the expected time it takes an A attacker to over-take honest miners if there is an e escrow period

Double Spending: Attack Duration in Closed Form

- Let $t(A, e)$ denote the expected time it takes an A attacker to over-take honest miners if there is an e escrow period
- Closed form expression:

$$t(A,e) = (1+e) + \left[\sum_{i=0}^{1+e} \left(\frac{i+1}{A-1}\right) \cdot \frac{(1+2e-i)!}{(1+e-i)!e!} \left(\frac{A}{1+A}\right)^{1+e-i} \left(\frac{1}{1+A}\right)^{1+e}\right].$$

Double Spending: Attack Duration in Closed Form

- Let $t(A, e)$ denote the expected time it takes an A attacker to over-take honest miners if there is an e escrow period
- Closed form expression:

$$t(A,e) = (1+e) + \left[\sum_{i=0}^{1+e} \left(\frac{i+1}{A-1}\right) \cdot \frac{(1+2e-i)!}{(1+e-i)!e!} \left(\frac{A}{1+A}\right)^{1+e-i} \left(\frac{1}{1+A}\right)^{1+e}\right].$$

- The attacker must wait for the honest chain to reach $1+e$ blocks due to the escrow condition no matter what — even if attacker's chain is much longer by then.
- What if the attacker's chain is *shorter* than the honest chain at time $1+e$? Call this difference in attacker and honest chain length the 'attacker deficit', i

Double Spending: Attack Duration in Closed Form

- Let $t(A, e)$ denote the expected time it takes an A attacker to over-take honest miners if there is an e escrow period
- Closed form expression:

$$t(A,e) = (1+e) + \left[\sum_{i=0}^{1+e} \left(\frac{i+1}{A-1}\right) \cdot \frac{(1+2e-i)!}{(1+e-i)!e!} \left(\frac{A}{1+A}\right)^{1+e-i} \left(\frac{1}{1+A}\right)^{1+e}\right].$$

- The attacker must wait for the honest chain to reach $1+e$ blocks due to the escrow condition no matter what — even if attacker's chain is much longer by then.
- What if the attacker's chain is *shorter* than the honest chain at time $1+e$? Call this difference in attacker and honest chain length the 'attacker deficit', i
  - The sum considers, for each possible attacker deficit at the end of the escrow period,
    - The expected time to overcome the attack deficit i: $\left(\frac{i+1}{A-1}\right)$
    - The probability of facing attack deficit i: $\frac{(1+2e-i)!}{(1+e-i)!e!} \left(\frac{A}{1+A}\right)^{1+e-i} \left(\frac{1}{1+A}\right)^{1+e}$

Double Spending Attack: Simulation Details I

Table 1, Panel A. Expected Duration of Attack (t)

	$e = 0$	$e = 1$	$e = 6$	$e = 10$	$e = 100$	$e = 1000$
$A = 1.05$	25.51	29.77	45.06	54.44	181.32	1,067.82
$A = 1.1$	13.02	15.42	24.48	30.35	125.81	1,004.04
$A = 1.2$	6.79	8.28	14.37	18.65	105.13	1,001.0
$A = 1.25$	5.54	6.86	12.41	16.44	102.79	1,001.0
$A = 1.33$	4.34	5.49	10.57	14.40	101.47	1,001.0
$A = 1.5$	3.08	4.07	8.77	12.49	101.03	1,001.0
$A = 2$	1.89	2.78	7.39	11.23	101.0	1,001.0
$A = 5$	1.12	2.06	7.00	11.00	101.0	1,001.0

Double Spending Attack: Simulation Details I

Table 1, Panel A. Expected Duration of Attack (t)

	$e = 0$	$e = 1$	$e = 6$	$e = 10$	$e = 100$	$e = 1000$
$A = 1.05$	25.51	29.77	45.06	54.44	181.32	1,067.82
$A = 1.1$	13.02	15.42	24.48	30.35	125.81	1,004.04
$A = 1.2$	6.79	8.28	14.37	18.65	105.13	1,001.0
$A = 1.25$	5.54	6.86	12.41	16.44	102.79	1,001.0
$A = 1.33$	4.34	5.49	10.57	14.40	101.47	1,001.0
$A = 1.5$	3.08	4.07	8.77	12.49	101.03	1,001.0
$A = 2$	1.89	2.78	7.39	11.23	101.0	1,001.0
$A = 5$	1.12	2.06	7.00	11.00	101.0	1,001.0

Double Spending Attack: Simulation Details II

Table 1, Panel B. Gross Cost of Attack (At)

	$e = 0$	$e = 1$	$e = 6$	$e = 10$	$e = 100$	$e = 1000$
$A = 1.05$	26.78	31.26	47.31	57.17	190.38	1,121.22
$A = 1.1$	14.32	16.96	26.92	33.39	138.39	1,104.45
$A = 1.2$	8.14	9.93	17.24	22.38	126.15	1,201.20
$A = 1.25$	6.93	8.57	15.51	20.55	128.49	1,251.25
$A = 1.33$	5.78	7.31	14.06	19.15	134.96	1,331.33
$A = 1.5$	4.62	6.11	13.15	18.73	151.54	1,501.5
$A = 2$	3.78	5.56	14.78	22.45	202.0	2,002.0
$A = 5$	5.59	10.29	35.01	55.00	505.0	5,005.0

Double Spending Attack: Simulation Details II

Table 1, Panel B. Gross Cost of Attack (At)

	$e = 0$	$e = 1$	$e = 6$	$e = 10$	$e = 100$	$e = 1000$
$A = 1.05$	26.78	31.26	47.31	57.17	190.38	1,121.22
$A = 1.1$	14.32	16.96	26.92	33.39	138.39	1,104.45
$A = 1.2$	8.14	9.93	17.24	22.38	126.15	1,201.20
$A = 1.25$	6.93	8.57	15.51	20.55	128.49	1,251.25
$A = 1.33$	5.78	7.31	14.06	19.15	134.96	1,331.33
$A = 1.5$	4.62	6.11	13.15	18.73	151.54	1,501.5
$A = 2$	3.78	5.56	14.78	22.45	202.0	2,002.0
$A = 5$	5.59	10.29	35.01	55.00	505.0	5,005.0

Double Spending Attack: Cases and Sensitivity

- Having discussed V_{attack} (as function of usefulness) and At (as function of escrow), now need to study κ. Approach: sensitivity analysis

Double Spending Attack: Cases and Sensitivity

- Having discussed V_{attack} (as function of usefulness) and At (as function of escrow), now need to study κ. Approach: sensitivity analysis
- A **base case,** $\kappa At = 16$. Corresponds to gross costs if escrow period $e = 6$ and attacker majority $A = 1.25$ (55%). Net costs if $\kappa = 1$ (frictions cancel block rewards).
- A **low friction case**, $\kappa At = 8$.
- An **expensive attack case**, $\kappa At = 150$. Corresponds to base case scenario but with much higher frictions ($\kappa = 10$). Also corresponds to one full day of block-compute-costs
- A **very expensive attack case** in which $\kappa At = 1000$. One full week of block-compute-costs

Securing Against an Attack: Base Case

Table 2. Cost to Secure Against Attack: Base Case Analysis

	Per-Block	Per-Day	Per-Year	Per-Transaction
Security Costs as % of Value Secured	6.25%	900%	328,500%	0.003%
To Secure:				
$1 thousand	$62.5 dollars	$9.0 thousand	$3.3 million	3.1 cents
$1 million	$62.5 thousand	$9.0 million	$3.3 billion	$31.3 dollars
$1 billion	$62.5 million	$9.0 billion	$3.3 trillion	$31.3 thousand
$100 billion	$6.3 billion	$900.0 billion	$328.5 trillion	$3.1 million

- Costs are high in absolute terms – follows directly from (3), rewritten as $\frac{p_{block}}{V_{attack}} \geq \frac{1}{\kappa A t}$
- Major difficulty: how costs scale. $100bn attack requires 4 times global GDP
- Looks more reasonable per transaction: but still scales poorly

Securing Against an Attack: Base Case

Table 2. Cost to Secure Against Attack: Base Case Analysis

	Per-Block	Per-Day	Per-Year	Per-Transaction
Security Costs as % of Value Secured	6.25%	900%	328,500%	0.003%
To Secure:				
$1 thousand	$62.5 dollars	$9.0 thousand	$3.3 million	3.1 cents
$1 million	$62.5 thousand	$9.0 million	$3.3 billion	$31.3 dollars
$1 billion	$62.5 million	$9.0 billion	$3.3 trillion	$31.3 thousand
$100 billion	$6.3 billion	$900.0 billion	$328.5 trillion	$3.1 million

- Costs are high in absolute terms – follows directly from (3), rewritten as $\frac{p_{block}}{V_{attack}} \geq \frac{1}{\kappa A t}$
- Major difficulty: how costs scale. $100bn attack requires 4 times global GDP
- Looks more reasonable per transaction: but still scales poorly

Securing Against an Attack: Base Case

Table 2. Cost to Secure Against Attack: Base Case Analysis

	Per-Block	Per-Day	Per-Year	Per-Transaction
Security Costs as % of Value Secured	6.25%	900%	328,500%	0.003%
To Secure:				
$1 thousand	$62.5 dollars	$9.0 thousand	$3.3 million	3.1 cents
$1 million	$62.5 thousand	$9.0 million	$3.3 billion	$31.3 dollars
$1 billion	$62.5 million	$9.0 billion	$3.3 trillion	$31.3 thousand
$100 billion	$6.3 billion	$900.0 billion	$328.5 trillion	$3.1 million

- Costs are high in absolute terms – follows directly from (3), rewritten as $\frac{p_{block}}{V_{attack}} \geq \frac{1}{\kappa A t}$
- Major difficulty: how costs scale. $100bn attack requires 4 times global GDP
- Looks more reasonable per transaction: but still scales poorly

Securing Against an Attack: Base Case

Table 2. Cost to Secure Against Attack: Base Case Analysis

	Per-Block	Per-Day	Per-Year	Per-Transaction
Security Costs as % of Value Secured	6.25%	900%	328,500%	0.003%
To Secure:				
$1 thousand	$62.5 dollars	$9.0 thousand	$3.3 million	3.1 cents
$1 million	$62.5 thousand	$9.0 million	$3.3 billion	$31.3 dollars
$1 billion	$62.5 million	$9.0 billion	$3.3 trillion	$31.3 thousand
$100 billion	$6.3 billion	$900.0 billion	$328.5 trillion	$3.1 million

- Costs are high in absolute terms – follows directly from (3), rewritten as $\frac{p_{block}}{V_{attack}} \geq \frac{1}{\kappa A t}$
- Major difficulty: how costs scale. $100bn attack requires 4 times global GDP
- Looks more reasonable per transaction: but still scales poorly

Securing Against an Attack: Sensitivity Analysis

Table 3, Panel B. Sensitivity Analysis: Cost to Secure Against $1B Attack.

Attack Scenarios	Per-Block	Per-Day	Per-Year	Per-Transaction
Base Case	$63 million	$9 billion	$3 trillion	$31 thousand
Low-Friction	$125 million	$18 billion	$7 trillion	$63 thousand
Expensive	$7 million	$960 million	$350 billion	$3 thousand
Very Expensive	$1 million	$144 million	$53 billion	$500 dollars

- Expensive and very expensive cases improve the picture by 1-2 orders of magnitude, but costs still very high
- To secure the system for large transactions (e.g. $1B attack) requires fees that are prohibitive for small transactions
- Even at a 1 week attack duration ($\kappa At = 1000$), require a per-transaction cost of $500 to keep Bitcoin secure up to $1Bn. 5% of global GDP for $100Bn.

Securing Against an Attack: Sensitivity Analysis

Table 3, Panel B. Sensitivity Analysis: Cost to Secure Against $1B Attack.

Attack Scenarios	Per-Block	Per-Day	Per-Year	Per-Transaction
Base Case	$63 million	$9 billion	$3 trillion	$31 thousand
Low-Friction	$125 million	$18 billion	$7 trillion	$63 thousand
Expensive	$7 million	$960 million	$350 billion	$3 thousand
Very Expensive	$1 million	$144 million	$53 billion	$500 dollars

- Expensive and very expensive cases improve the picture by 1-2 orders of magnitude, but costs still very high
- To secure the system for large transactions (e.g. $1B attack) requires fees that are prohibitive for small transactions
- Even at a 1 week attack duration ($\kappa At = 1000$), require a per-transaction cost of $500 to keep Bitcoin secure up to $1Bn. 5% of global GDP for $100Bn.

Securing Against an Attack: Sensitivity Analysis

Table 3, Panel B. Sensitivity Analysis: Cost to Secure Against $1B Attack.

Attack Scenarios	Per-Block	Per-Day	Per-Year	Per-Transaction
Base Case	$63 million	$9 billion	$3 trillion	$31 thousand
Low-Friction	$125 million	$18 billion	$7 trillion	$63 thousand
Expensive	$7 million	$960 million	$350 billion	$3 thousand
Very Expensive	$1 million	$144 million	$53 billion	$500 dollars

- Expensive and very expensive cases improve the picture by 1-2 orders of magnitude, but costs still very high
- To secure the system for large transactions (e.g. $1B attack) requires fees that are prohibitive for small transactions
- Even at a 1 week attack duration ($\kappa At = 1000$), require a per-transaction cost of $500 to keep Bitcoin secure up to $1Bn. 5% of global GDP for $100Bn.

Double Spending Attack: Takeaways

$$p_{block} > \frac{V_{attack}}{\kappa A t}$$

- Some takeaways from the double-spending simulations:

Double Spending Attack: Takeaways

$$p_{block} > \frac{V_{attack}}{\kappa A t}$$

- Some takeaways from the double-spending simulations:
- Consistent with the modest early use cases of Bitcoin (computer parts, silk road, online gambling) — if double-spending worth \$1k, then implicit cost per tx just \$0.03

Double Spending Attack: Takeaways

$$p_{block} > \frac{V_{attack}}{\kappa A t}$$

- Some takeaways from the double-spending simulations:
- Consistent with the modest early use cases of Bitcoin (computer parts, silk road, online gambling) — if double-spending worth \$1k, then implicit cost per tx just \$0.03
- Consistent with larger-scale black-market uses of Bitcoin — users willing to pay the high costs

Double Spending Attack: Takeaways

$$p_{block} > \frac{V_{attack}}{\kappa A t}$$

- Some takeaways from the double-spending simulations:
- Consistent with the modest early use cases of Bitcoin (computer parts, silk road, online gambling) — if double-spending worth \$1k, then implicit cost per tx just \$0.03
- Consistent with larger-scale black-market uses of Bitcoin — users willing to pay the high costs
- Casts doubt on Bitcoin as major component of mainstream global financial system (too expensive!)

Double Spending Attack: Takeaways

$$p_{block} > \frac{V_{attack}}{\kappa A t}$$

- Some takeaways from the double-spending simulations:
- Consistent with the modest early use cases of Bitcoin (computer parts, silk road, online gambling) — if double-spending worth \$1k, then implicit cost per tx just \$0.03
- Consistent with larger-scale black-market uses of Bitcoin — users willing to pay the high costs
- Casts doubt on Bitcoin as major component of mainstream global financial system (too expensive!)
- **For the system to be secure for large transactions requires implicit tax rates that render it unusable for small ones**

Double Spending Attack: Takeaways

$$p_{block} > \frac{V_{attack}}{\kappa A t}$$

- Some takeaways from the double-spending simulations:
- Consistent with the modest early use cases of Bitcoin (computer parts, silk road, online gambling) — if double-spending worth $1k, then implicit cost per tx just $0.03
- Consistent with larger-scale black-market uses of Bitcoin — users willing to pay the high costs
- Casts doubt on Bitcoin as major component of mainstream global financial system (too expensive!)
- **For the system to be secure for large transactions requires implicit tax rates that render it unusable for small ones**
- Surprise to CS community: that escrow period isn't more protective
  - Intuition I: attacker earns block rewards while waiting for escrow to clear
  - Intuition II: attack stakes grow if Bitcoin gets more economically useful / important

Attack II: Sabotage

- Obvious response: double spending would be "noticed"
- Cause decline in value of Bitcoin, which attacker needs to hold
- Bitcoin Wiki classifies majority attack "Probably Not a Problem" for this reason

Attack II: Sabotage

- Obvious response: double spending would be "noticed"
- Cause decline in value of Bitcoin, which attacker needs to hold
- Bitcoin Wiki classifies majority attack "Probably Not a Problem" for this reason
- Formally: suppose Bitcoin value declines by proportion Δ_{attack}
- Constraint is now:

$$p_{block} > \frac{(1 - \Delta_{attack})}{At(\kappa + \Delta_{attack})} V_{attack}$$

Attack II: Sabotage

- Obvious response: double spending would be "noticed"
- Cause decline in value of Bitcoin, which attacker needs to hold
- Bitcoin Wiki classifies majority attack "Probably Not a Problem" for this reason
- Formally: suppose Bitcoin value declines by proportion Δ_{attack}
- Constraint is now:

$$p_{block} > \frac{(1 - \Delta_{attack})}{At(\kappa + \Delta_{attack})} V_{attack}$$

- If Δ_{attack} large enough, then indeed deter double spending ...

Attack II: Sabotage

- Obvious response: double spending would be "noticed"
- Cause decline in value of Bitcoin, which attacker needs to hold
- Bitcoin Wiki classifies majority attack "Probably Not a Problem" for this reason
- Formally: suppose Bitcoin value declines by proportion Δ_{attack}
- Constraint is now:

$$p_{block} > \frac{(1 - \Delta_{attack})}{At(\kappa + \Delta_{attack})} V_{attack}$$

- If Δ_{attack} large enough, then indeed deter double spending ...
- However, "pick your poison":

Attack II: Sabotage

- Obvious response: double spending would be "noticed"
- Cause decline in value of Bitcoin, which attacker needs to hold
- Bitcoin Wiki classifies majority attack "Probably Not a Problem" for this reason
- Formally: suppose Bitcoin value declines by proportion Δ_{attack}
- Constraint is now:

$$p_{block} > \frac{(1 - \Delta_{attack})}{At(\kappa + \Delta_{attack})} V_{attack}$$

- If Δ_{attack} large enough, then indeed deter double spending ...
- However, "pick your poison":
  - Need to concede possibility of sabotage/collapse

Attack II: Sabotage

- Obvious response: double spending would be "noticed"
- Cause decline in value of Bitcoin, which attacker needs to hold
- Bitcoin Wiki classifies majority attack "Probably Not a Problem" for this reason
- Formally: suppose Bitcoin value declines by proportion Δ_{attack}
- Constraint is now:

$$p_{block} > \frac{(1 - \Delta_{attack})}{At(\kappa + \Delta_{attack})} V_{attack}$$

- If Δ_{attack} large enough, then indeed deter double spending ...
- However, "pick your poison":
  - Need to concede possibility of sabotage/collapse
  - Then should worry about attacker motivated by sabotage per se: $V_{sabotage}$

Attack II: Sabotage

- Obvious response: double spending would be "noticed"
- Cause decline in value of Bitcoin, which attacker needs to hold
- Bitcoin Wiki classifies majority attack "Probably Not a Problem" for this reason
- Formally: suppose Bitcoin value declines by proportion Δ_{attack}
- Constraint is now:

$$p_{block} > \frac{(1 - \Delta_{attack})}{At(\kappa + \Delta_{attack})} V_{attack}$$

- If Δ_{attack} large enough, then indeed deter double spending ...
- However, "pick your poison":
  - Need to concede possibility of sabotage/collapse
  - Then should worry about attacker motivated by sabotage per se: $V_{sabotage}$
  - **Either: high implicit tax rates or risk of collapse**

Attack II: Sabotage

- How big is $V_{sabotage}$?

Attack II: Sabotage

- How big is $V_{sabotage}$?
- Hard to say of course
- Easy to imagine magnitudes already large
- Would be larger still if Bitcoin becomes more integrated into global financial system

Attack II: Sabotage

- How big is $V_{sabotage}$?
- Hard to say of course
- Easy to imagine magnitudes already large
- Would be larger still if Bitcoin becomes more integrated into global financial system
- Futures markets
  - CME: $2bn of open interest (April 2022)
  - Crypto Exchanges: $20bn of open interest
  - (Note: was just $160 million at time of June 2018 draft!)
- Bitcoin market capitalization: as high as $1 trillion
  - (Peter Thiel prediction: $100 trillion)
- These figures give some sense of magnitudes for economic harm a bad actor could cause by sabotage

Overview of the Talk

- Overview: What is Nakamoto Blockchain
- Nakamoto Blockchain: A Critique in 3 Equations
  - Rent-Seeking Competition (Miners)
  - Incentive Compatibility (Majority Attack)
  - Economic Constraint on the Blockchain: Flow vs. Stock
- Majority Attack Scenarios
  - Double Spending
  - Sabotage
- **Blockchain-Specific Mining Technology**
  - **A Softer Constraint: Stock vs. Stock**
  - **Collapse Scenarios**

Sabotage and Blockchain-Specific Capital

- Nakamoto (2008) envisioned ordinary computers ("one-CPU-one-vote")
- Since 2013, Bitcoin dominated by specialized equipment
  - ASICs = Application Specific Integrated Circuits
  - Not just a bit more efficient ... factor of 10,000x or more
- As emphasized, if capital is specialized, and attack causes a collapse of the trust, then attacker cost model needs to be modified
  - In addition to charging attacker the flow cost of $(1 + \kappa)At \cdot c$
  - Also need to charge attacker for decline in value of specialized capital
- Makes cost more like a stock than a flow — thus blockchain much more secure
- But this security rests on the fragile precipice of specific capital and vulnerability to sabotage

Antminer

- Cost per machine
  - S19 Pro: $3769 (March 2021)
  - S19 Pro: $7700 (May 2022)
- Mining power: 104-110 TH/s
- Cost to match the Bitcoin hash rate:
  - Mar 2021: $5bn
  - May 2022: $15bn

Note: The numbers are based on data from March 2021 and 2022. Data from shop.bitmain.com.

Amazon Web Services

- AWS Total computation equipment in 2021: $65 bn
- Assume ASIC machines are 10000 times more cost effective than AWS machines (conservative)
- Devoting all AWS to Bitcoin mining will get about .05% of total network hash rate

Note: The numbers are based on data from early 2022. Data of Amazon AWS total PP&E and potential equipment lease are obtained from Amazon 10-K. The cost/efficiency ratio is a conservative estimate based on the data of the hash rate of non-specific mining hardware obtained from Bitcoin Wiki.

Cost to Secure Against Sabotage, Derivation

- Write per-unit-time compute cost as $c = rC + \eta$. Rental cost of capital plus variable cost of electricity. Honest mining equilibrium (1) can be written as:

$$N^*(rC + \eta) = p_{block}.$$

- Outside attacker needs N^*C of capital. Insider needs $\frac{N^*C}{2}$.

Cost to Secure Against Sabotage, Derivation

- Write per-unit-time compute cost as $c = rC + \eta$. Rental cost of capital plus variable cost of electricity. Honest mining equilibrium (1) can be written as:

$$N^*(rC + \eta) = p_{block}.$$

- Outside attacker needs N^*C of capital. Insider needs $\frac{N^*C}{2}$.
- We can compute these as a function of p_{block}. Let $\mu = \frac{rC}{rC+\eta}$ denote the capital share of mining. Then:

$$N^*C = \frac{\mu p_{block}}{r}.$$

- Note: if we use $\mu = 0.4$ (De Vries, 2018), $r = 50\%$ per year (depreciation, risk), and $p_{block} = \$15B$ per year, then formula implies $N^*C = \$12B$.

Cost to Secure Against Sabotage, Derivation

- Write per-unit-time compute cost as $c = rC + \eta$. Rental cost of capital plus variable cost of electricity. Honest mining equilibrium (1) can be written as:

$$N^*(rC + \eta) = p_{block}.$$

- Outside attacker needs N^*C of capital. Insider needs $\frac{N^*C}{2}$.
- We can compute these as a function of p_{block}. Let $\mu = \frac{rC}{rC+\eta}$ denote the capital share of mining. Then:

$$N^*C = \frac{\mu p_{block}}{r}.$$

- Note: if we use $\mu = 0.4$ (De Vries, 2018), $r = 50\%$ per year (depreciation, risk), and $p_{block} = \$15B$ per year, then formula implies $N^*C = \$12B$.
- IC constraint to secure against outsider sabotage is approximated by:

$$N^*C > V_{sabotage} \implies \frac{\mu p_{block}}{r} > V_{sabotage}. \tag{8}$$

- An insider attacker only requires $\geq \frac{N^*C}{2}$ worth of capital, for analogous result

Cost to Secure Against Sabotage

Table 4. Cost Per (8) to Secure Against Sabotage

A. Cost to Secure Against Outsider Sabotage

Value of Sabotage	Annual Miner Payments	Specialized Capital Stock
$1 billion	$1.25 billion	$1 billion
$10 billion	$12.5 billion	$10 billion
$100 billion	$125 billion	$100 billion
$1 trillion	$1.25 trillion	$1 trillion

B. Cost to Secure Against Insider Sabotage

Value of Sabotage	Annual Miner Payments	Specialized Capital Stock
$1 billion	$2.5 billion	$2 billion
$10 billion	$25 billion	$20 billion
$100 billion	$250 billion	$200 billion
$1 trillion	$2.5 trillion	$2 trillion

Cost Per (8) to Secure Against Sabotage

Table 4. Cost Per (8) to Secure Against Sabotage

A. Cost to Secure Against Outsider Sabotage

Value of Sabotage	Annual Miner Payments	Specialized Capital Stock
$1 billion	$1.25 billion	$1 billion
$10 billion	$12.5 billion	$10 billion
$100 billion	$125 billion	$100 billion
$1 trillion	$1.25 trillion	$1 trillion

B. Cost to Secure Against Insider Sabotage

Value of Sabotage	Annual Miner Payments	Specialized Capital Stock
$1 billion	$2.5 billion	$2 billion
$10 billion	$25 billion	$20 billion
$100 billion	$250 billion	$200 billion
$1 trillion	$2.5 trillion	$2 trillion

Discussion of Sabotage Analysis

- If one concedes that a majority attack would effectively be a sabotage that causes the trust model to collapse, then Bitcoin is *much* more secure than we found earlier
- However, still several important concerns

1. Security model still scales linearly. [Would not trust Federal Reserve or Fort Knox with this security model!]
2. Analysis rests on the nature of the chip market
3. Analysis premised on assumption that attack causes significant collapse in value –> this itself is a vulnerability

- These concerns point to collapse scenarios

Collapse Scenarios

- Suppose, for purpose of discussion
  - Bitcoin blockchain *does not* satisfy (2): $\kappa At \cdot N^* c > V_{attack}$
  - Bitcoin blockchain *does* satisfy (2'): $N^* C > V_{attack}$

Collapse Scenarios

- Suppose, for purpose of discussion
  - Bitcoin blockchain *does not* satisfy (2): $\kappa At \cdot N^* c > V_{attack}$
  - Bitcoin blockchain *does* satisfy (2'): $N^* C > V_{attack}$
- Analysis then suggests 3 possible scenarios that could precipitate collapse:

Collapse Scenarios

- Suppose, for purpose of discussion
  - Bitcoin blockchain *does not* satisfy (2): $\kappa At \cdot N^* c > V_{attack}$
  - Bitcoin blockchain *does* satisfy (2'): $N^* C > V_{attack}$
- Analysis then suggests 3 possible scenarios that could precipitate collapse:

1. Cheap-enough specialized ASIC chips
   - Then cost becomes a flow not a stock

Collapse Scenarios

- Suppose, for purpose of discussion
  - Bitcoin blockchain *does not* satisfy (2): $\kappa At \cdot N^* c > V_{attack}$
  - Bitcoin blockchain *does* satisfy (2'): $N^* C > V_{attack}$
- Analysis then suggests 3 possible scenarios that could precipitate collapse:

1. Cheap-enough specialized ASIC chips
   - Then cost becomes a flow not a stock
2. If Bitcoin value falls (for other reasons):
   - Glut of chips relative to amount needed for mining equilibrium (1) -> flow

Collapse Scenarios

- Suppose, for purpose of discussion
  - Bitcoin blockchain *does not* satisfy (2): $\kappa At \cdot N^* c > V_{attack}$
  - Bitcoin blockchain *does* satisfy (2'): $N^* C > V_{attack}$
- Analysis then suggests 3 possible scenarios that could precipitate collapse:

1. Cheap-enough specialized ASIC chips
   - Then cost becomes a flow not a stock
2. If Bitcoin value falls (for other reasons):
   - Glut of chips relative to amount needed for mining equilibrium (1) -> flow
3. Bitcoin grows in economic importance (rel. to cost)
   - Futures markets grow (short-seller attack)
   - Bitcoin becomes more fully integrated in global financial system (double-spend or saboteur)

Attack Scenario 1. Cheap-enough Specialized Chips

- Suppose there are previous-generation ASIC chips that are not economically efficient for mining, but are powerful enough for the purpose of attack and exist in large quantity
  - Formally, suppose per-unit-time electricity cost is $\eta' > c$. So in honest mining equilibrium, old chips are not economical to use even if the chips themselves are free.
  - If there are $\geq N^*$ compute units of old chips, then attacker can attack at flow cost of $N^*\eta'$.

Attack Scenario 1. Cheap-enough Specialized Chips

- Suppose there are previous-generation ASIC chips that are not economically efficient for mining, but are powerful enough for the purpose of attack and exist in large quantity
  - Formally, suppose per-unit-time electricity cost is $\eta' > c$. So in honest mining equilibrium, old chips are not economical to use even if the chips themselves are free.
  - If there are $\geq N^*$ compute units of old chips, then attacker can attack at flow cost of $N^*\eta'$.
- Even if $\eta' = 10c$, cost of attack in base case is $16 \cdot 10 \cdot p_{block} = \$40$ million at $p_{block} = \$250k$. Much cheaper than N^*C.

Attack Scenario 1. Cheap-enough Specialized Chips

- Suppose there are previous-generation ASIC chips that are not economically efficient for mining, but are powerful enough for the purpose of attack and exist in large quantity
  - Formally, suppose per-unit-time electricity cost is $\eta' > c$. So in honest mining equilibrium, old chips are not economical to use even if the chips themselves are free.
  - If there are $\geq N^*$ compute units of old chips, then attacker can attack at flow cost of $N^*\eta'$.
- Even if $\eta' = 10c$, cost of attack in base case is $16 \cdot 10 \cdot p_{block} = \$40$ million at $p_{block} = \$250k$. Much cheaper than N^*C.
- Currently no reason to think $\geq N^*$ compute units of old chips exist
  - Both quantity and quality have been growing dramatically
- But ASIC market continues to mature -> this could change.

Attack Scenario 2. Sufficient Fall in Mining Rewards

- Recall $N^*(rC + \eta) = p_{block}$ and $\mu :=$the capital share of mining cost.
- If p_{block} falls to $\alpha \cdot p_{block}$, with $\alpha < (1 - \mu)$, then $N^*\eta > \alpha \cdot p_{block}$ and capital will be "mothballed".

Attack Scenario 2. Sufficient Fall in Mining Rewards

- Recall $N^*(rC + \eta) = p_{block}$ and $\mu :=$the capital share of mining cost.
- If p_{block} falls to $\alpha \cdot p_{block}$, with $\alpha < (1 - \mu)$, then $N^*\eta > \alpha \cdot p_{block}$ and capital will be "mothballed".
- Example: if $\mu = 0.4$, then some capital is mothballed upon 30% decline, and 50% of capital is mothballed upon 70% decline
- If enough capital is mothballed for a sufficiently long period of time, this would seem to raise the vulnerability to attack

Attack Scenario 2. Sufficient Fall in Mining Rewards

- Recall $N^*(rC + \eta) = p_{block}$ and $\mu :=$the capital share of mining cost.
- If p_{block} falls to $\alpha \cdot p_{block}$, with $\alpha < (1 - \mu)$, then $N^*\eta > \alpha \cdot p_{block}$ and capital will be "mothballed".
- Example: if $\mu = 0.4$, then some capital is mothballed upon 30% decline, and 50% of capital is mothballed upon 70% decline
- If enough capital is mothballed for a sufficiently long period of time, this would seem to raise the vulnerability to attack
- Additionally, Bitcoin halvings will decrease p_{block} over time.
  - By 2032, reward is <1 Bitcoin
  - By 2044, reward is <0.1 Bitcoin
- Hence: either Bitcoin value must grow significantly, transaction costs must grow significantly, or there will be significant mothballed capital -> vulnerability

Attack Scenario 3. Bitcoin Grows in Economic Importance (Relative to Cost)

- Previous two scenarios identify conditions under which the cost of attack changes from a stock cost to a flow cost

Attack Scenario 3. Bitcoin Grows in Economic Importance (Relative to Cost)

- Previous two scenarios identify conditions under which the cost of attack changes from a stock cost to a flow cost
- The other possibility: Bitcoin grows in economic importance, enough to tempt a saboteur
  - That is, $V_{attack} > N^* C$.
- Speculatively, this seems most likely to occur if Bitcoin becomes more fully integrated into the global financial system.
  - $12bn is small in the scheme of global finance

Examples of 51% Attacks

Name	Hash function	Date of First Attack	Amount Stolen
Verge	Scrypt, X17, Lyra2rev2, Myr-groestl, Blake2s	4/4/2018	$2,800,000
		2/15/2021	Unknown
Monacoin	Lyra2rev2	5/13/2018	$90,000
Bitcoin Gold	Equihash	5/16/2018	$18,000,000
		1/23/2020	$72,000
Litecoin Cash	SHA-256	5/30/2018	Unknown
Zencash	Equihash	6/2/2018	$700,000
Vertcoin	Lyra2rev2	10/12/2018	$100,000
Bitcoin SV	SHA-256	8/3/2021	Unknown
Ethereum Classic	Ethash	1/5/2019	$1,100,000
		8/1/2020	$7,280,000

Sources: Bloomberg, Coindesk, Bitcoinist, CCN, Cointelegraph, bitquery, GitHub Gist and Medium. The hash functions listed here are the hash functions at the time of the attack. Often there is an ambiguity of whether several block reorganizations should be considered as 1 attack or several attacks. Because of this, only the date of the first attack/reorganization is mentioned.

Examples of 51% Attacks

Name	Hash function	Date of First Attack	Amount Stolen
Verge	Scrypt, X17, Lyra2rev2, Myr-groestl, Blake2s	4/4/2018	$2,800,000
		2/15/2021	Unknown
Monacoin	Lyra2rev2	5/13/2018	$90,000
Bitcoin Gold	**Equihash**	**5/16/2018**	**$18,000,000**
		1/23/2020	$72,000
Litecoin Cash	SHA-256	5/30/2018	Unknown
Zencash	Equihash	6/2/2018	$700,000
Vertcoin	Lyra2rev2	10/12/2018	$100,000
Bitcoin SV	SHA-256	8/3/2021	Unknown
Ethereum Classic	Ethash	1/5/2019	$1,100,000
		8/1/2020	$7,280,000

Sources: Bloomberg, Coindesk, Bitcoinist, CCN, Cointelegraph, bitquery, GitHub Gist and Medium. The hash functions listed here are the hash functions at the time of the attack. Often there is an ambiguity of whether several block reorganizations should be considered as 1 attack or several attacks. Because of this, only the date of the first attack/reorganization is mentioned.

Conclusion: Summary

- Anonymous, decentralized trust enabled by Nakamoto (2008) blockchain: *ingenious but expensive*

Conclusion: Summary

- Anonymous, decentralized trust enabled by Nakamoto (2008) blockchain: *ingenious but expensive*
- Eq. (3): for trust to be meaningful, flow cost of running the blockchain > one-shot value of attacking it

Conclusion: Summary

- Anonymous, decentralized trust enabled by Nakamoto (2008) blockchain: *ingenious but expensive*
- Eq. (3): for trust to be meaningful, flow cost of running the blockchain > one-shot value of attacking it
  - To prevent double spending: payments to miners must be large relative to the highest-value uses of Bitcoin

Conclusion: Summary

- Anonymous, decentralized trust enabled by Nakamoto (2008) blockchain: *ingenious but expensive*
- Eq. (3): for trust to be meaningful, flow cost of running the blockchain > one-shot value of attacking it
  - To prevent double spending: payments to miners must be large relative to the highest-value uses of Bitcoin
  - Like a large implicit tax

Conclusion: Summary

- Anonymous, decentralized trust enabled by Nakamoto (2008) blockchain: *ingenious but expensive*
- Eq. (3): for trust to be meaningful, flow cost of running the blockchain > one-shot value of attacking it
  - To prevent double spending: payments to miners must be large relative to the highest-value uses of Bitcoin
  - Like a large implicit tax
- Argument that attack costs more than this flow cost requires one to concede both
  1. Security relies on use of scarce, specialized chips (contra Nakamoto ideal)
  2. Vulnerable to sabotage, collapse ("pick your poison")

Conclusion: Summary

- Anonymous, decentralized trust enabled by Nakamoto (2008) blockchain: *ingenious but expensive*
- Eq. (3): for trust to be meaningful, flow cost of running the blockchain > one-shot value of attacking it
  - To prevent double spending: payments to miners must be large relative to the highest-value uses of Bitcoin
  - Like a large implicit tax
- Argument that attack costs more than this flow cost requires one to concede both
  1. Security relies on use of scarce, specialized chips (contra Nakamoto ideal)
  2. Vulnerable to sabotage, collapse ("pick your poison")
- The analysis then points to specific collapse scenarios

Conclusion: Summary

- Anonymous, decentralized trust enabled by Nakamoto (2008) blockchain: *ingenious but expensive*
- Eq. (3): for trust to be meaningful, flow cost of running the blockchain > one-shot value of attacking it
  - To prevent double spending: payments to miners must be large relative to the highest-value uses of Bitcoin
  - Like a large implicit tax
- Argument that attack costs more than this flow cost requires one to concede both
  1. Security relies on use of scarce, specialized chips (contra Nakamoto ideal)
  2. Vulnerable to sabotage, collapse ("pick your poison")
- The analysis then points to specific collapse scenarios
- Overall message: there are intrinsic economic limits to how economically important Bitcoin can become. If it gets important enough, it will be attacked. (Unless payments to miners grow even higher)

Conclusion: Remark

- Emphasize: model consistent with earliest uses of Bitcoin and blockchain: hobbyists and black market
  - Black market = willing to pay high implicit fees
- Skepticism:
  - Bitcoin as "store of value" akin to gold
  - Bitcoin as a major component of the global financial system
  - Use of Nakamoto blockchain by businesses, governments

Conclusion: Remark

- Emphasize: model consistent with earliest uses of Bitcoin and blockchain: hobbyists and black market
  - Black market = willing to pay high implicit fees
- Skepticism:
  - Bitcoin as "store of value" akin to gold
  - Bitcoin as a major component of the global financial system
  - Use of Nakamoto blockchain by businesses, governments
- Also emphasize: not skeptical of use of distributed databases more broadly

Conclusion: Remark

- Emphasize: model consistent with earliest uses of Bitcoin and blockchain: hobbyists and black market
  - Black market = willing to pay high implicit fees
- Skepticism:
  - Bitcoin as "store of value" akin to gold
  - Bitcoin as a major component of the global financial system
  - Use of Nakamoto blockchain by businesses, governments
- Also emphasize: not skeptical of use of distributed databases more broadly
- What this paper highlights is that it is exactly the aspect of Bitcoin and Nakamoto (2008) that is so innovative relative to traditional distributed databases — *the anonymous, decentralized trust that emerges from proof-of-work* — that also may make it so economically limited

Conclusion: Open Question I

- Open question: are there other ways to generate anonymous, decentralized trust that make this paper's arguments less constraining?
  - Characterization theorems of Leshno and Strack (2020) and Chen, Papdimitirou and Roughgarden (2019) provide some useful constraints on the problem space
  - Axioms that relate to strict interpretations of anonymity and decentralization (invariance to name changes, free entry, collusion proof) -> Nakamoto compensation scheme
  - Hence options are (i) modify Nakamoto without touching compensation scheme per se, or (ii) relax anonymity or decentralization

Conclusion: Open Question I

- Open question: are there other ways to generate anonymous, decentralized trust that make this paper's arguments less constraining?
  - Characterization theorems of Leshno and Strack (2020) and Chen, Papdimitirou and Roughgarden (2019) provide some useful constraints on the problem space
  - Axioms that relate to strict interpretations of anonymity and decentralization (invariance to name changes, free entry, collusion proof) -> Nakamoto compensation scheme
  - Hence options are (i) modify Nakamoto without touching compensation scheme per se, or (ii) relax anonymity or decentralization
- Interesting in this regard: "proof of stake"
  - Usual motivation: reduce mining expense and environmental harm (Bitcoin is 0.3-0.8% of *global* electricity consumption)
  - Environmental issue is orthogonal to the concerns raised in this paper. Just conceptualize c as per-block opportunity cost of stake (Gans and Gandal, 2019)
  - But: stakes have *memory*. This may open up new possibilities for thwarting attacks.

Conclusion: Open Question I

- Open question: are there other ways to generate anonymous, decentralized trust that make this paper's arguments less constraining?
  - Characterization theorems of Leshno and Strack (2020) and Chen, Papdimitirou and Roughgarden (2019) provide some useful constraints on the problem space
  - Axioms that relate to strict interpretations of anonymity and decentralization (invariance to name changes, free entry, collusion proof) -> Nakamoto compensation scheme
  - Hence options are (i) modify Nakamoto without touching compensation scheme per se, or (ii) relax anonymity or decentralization
- Interesting in this regard: "proof of stake"
  - Usual motivation: reduce mining expense and environmental harm (Bitcoin is 0.3-0.8% of *global* electricity consumption)
  - Environmental issue is orthogonal to the concerns raised in this paper. Just conceptualize c as per-block opportunity cost of stake (Gans and Gandal, 2019)
  - But: stakes have *memory*. This may open up new possibilities for thwarting attacks.
- Very active research area ... open question what exactly is possible "in between" Nakamoto trust and traditional trust

Conclusion: Open Question II

- Computer scientists unimpressed with "private blockchain" / "distributed ledger"
  - "Just a database"
  - Nothing intellectually new from a CS perspective

Conclusion: Open Question II

- Computer scientists unimpressed with "private blockchain" / "distributed ledger"
  - "Just a database"
  - Nothing intellectually new from a CS perspective
- Open question: is there anything *economically* novel that emerges from this particular form of database?
  - Features: append-only, secure timestamps, appends pushed to all parties, pre-specified permissions as to who can do what, etc.
  - But with trust ultimately coming from traditional sources: rule of law, relationships, reputations, etc.
  - Goldman Sachs: "New Technology of Trust"
  - Amazon Blockchain just announced
  - BIS, JPMorgan, Walmart
  - NFTs from the NBA

Conclusion: Open Question III

- There is clearly a lot of cultural, intellectual and financial excitement about Nakamoto's novel form of trust, and decentralization more broadly

Conclusion: Open Question III

- There is clearly a lot of cultural, intellectual and financial excitement about Nakamoto's novel form of trust, and decentralization more broadly
- Yet, most volume appears to involve cryptocurrency exchanges — centralized, trusted, financial intermediaries! (Makarov and Schoar, 2021)
- Open question: how do we make sense of this?

Conclusion: Open Question III

- There is clearly a lot of cultural, intellectual and financial excitement about Nakamoto's novel form of trust, and decentralization more broadly
- Yet, most volume appears to involve cryptocurrency exchanges — centralized, trusted, financial intermediaries! (Makarov and Schoar, 2021)
- Open question: how do we make sense of this?
- Perhaps the key distinction is between *users* of Nakamoto's novel form of trust and *speculators* about its importance — the latter of whom are perfectly happy to transact via traditional financial intermediaries.
- If so, then this paper suggests cause for concern for such speculators, if their speculation is premised on eventual high economic usefulness
  - (Though I caution it is not possible to draw a direct line from this paper to an appropriate valuation. See Athey et al (2016) for an early effort to model valuation as a function of usefulness).

Conclusion: Responses to June 2018 Draft

1. Community
2. Rule of Law (ironic)
3. Counterattacks
4. Modification to Nakamoto I: Increase Throughput
5. Modification to Nakamoto II: Tweak Longest-Chain Convention
6. Proof of Stake