The Economic Limits of Bitcoin and the Blockchain

Eric Budish
Chicago Booth

Atlanta Fed Financial Markets Conference
May 2019

Overview of the Argument

Overview of the Argument

- Satoshi Nakamoto's innovation: anonymous, decentralized trust in a dataset, from "proof-of-work"

Overview of the Argument

- Satoshi Nakamoto's innovation: anonymous, decentralized trust in a dataset, from "proof-of-work"
- Amount of computational work must simultaneously satisfy:

Overview of the Argument

- Satoshi Nakamoto's innovation: anonymous, decentralized trust in a dataset, from "proof-of-work"
- Amount of computational work must simultaneously satisfy:
  - (1) zero-profits condition: for blockchain "miners"

Overview of the Argument

- Satoshi Nakamoto's innovation: anonymous, decentralized trust in a dataset, from "proof-of-work"
- Amount of computational work must simultaneously satisfy:
  - (1) zero-profits condition: for blockchain "miners"
  - (2) incentive compatibility condition: deter "majority attack"

Overview of the Argument

- Satoshi Nakamoto's innovation: anonymous, decentralized trust in a dataset, from "proof-of-work"
- 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:

Overview of the Argument

- Satoshi Nakamoto's innovation: anonymous, decentralized trust in a dataset, from "proof-of-work"
- 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

Overview of the Argument

- Satoshi Nakamoto's innovation: anonymous, decentralized trust in a dataset, from "proof-of-work"
- 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.

Overview of the Argument

- Satoshi Nakamoto's innovation: anonymous, decentralized trust in a dataset, from "proof-of-work"
- 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.
- Way out (i.e., why hasn't Bitcoin already been attacked):

Overview of the Argument

- Satoshi Nakamoto's innovation: anonymous, decentralized trust in a dataset, from "proof-of-work"
- 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.
- Way out (i.e., why hasn't Bitcoin already been attacked):
  - (i) mining technology is specialized/non-repurposable, <u>and</u>
  - (ii) any majority attack is a "sabotage", causes collapse

Overview of the Argument

- Satoshi Nakamoto's innovation: anonymous, decentralized trust in a dataset, from "proof-of-work"
- 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.
- Way out (i.e., why hasn't Bitcoin already been attacked):
  - (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; analysis points to specific collapse scenarios

Overview of the Argument

- Satoshi Nakamoto's innovation: anonymous, decentralized trust in a dataset, from "proof-of-work"
- 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.
- Way out (i.e., why hasn't Bitcoin already been attacked):
  - (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; analysis points to specific collapse scenarios
- Overall take: ingenious, but economically limited.

What is Nakamoto Blockchain (1/3)

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

What is Nakamoto Blockchain (1/3)

- **Transaction:** sender, receiver, amount, signature
- **Signature**:
  - Proves sender's identity
  - Encodes transaction details (amount, recipient)
  - Standard cryptography techniques
- Imagine transactions on a google spreadsheet
  - Signature: only I can add transactions in which I send money
  - But:
    - I can send money I don't have
    - I can send money I do have but to multiple parties at the same time
    - I can delete previous transactions (mine or others')

What is Nakamoto Blockchain (1/3)

- **Transaction:** sender, receiver, amount, signature
- **Signature**:
  - Proves sender's identity
  - Encodes transaction details (amount, recipient)
  - Standard cryptography techniques
- Imagine transactions on a google spreadsheet
  - Signature: only I can add transactions in which I send money
  - But:
    - I can send money I don't have
    - I can send money I do have but to multiple parties at the same time
    - I can delete previous transactions (mine 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 (2/3)

Nakamoto (2008) Blockchain Innovation

What is Nakamoto Blockchain (2/3)

Nakamoto (2008) Blockchain Innovation

- Users submit transactions to a pending transactions list

What is Nakamoto Blockchain (2/3)

Nakamoto (2008) Blockchain Innovation

- Users submit transactions to a pending transactions list
- Anonymous, decentralized mass of compute power ("miners") competes to validate the next block of transactions

What is Nakamoto Blockchain (2/3)

Nakamoto (2008) Blockchain Innovation

- Users submit transactions to a pending transactions list
- Anonymous, decentralized mass of compute power ("miners") competes to validate the next block of transactions
  - Each new block "chains" to previous block
  - Validity: each transaction must (i) be properly signed, (ii) be funded given previous blocks, (iii) not contradict other transactions in this block

What is Nakamoto Blockchain (2/3)

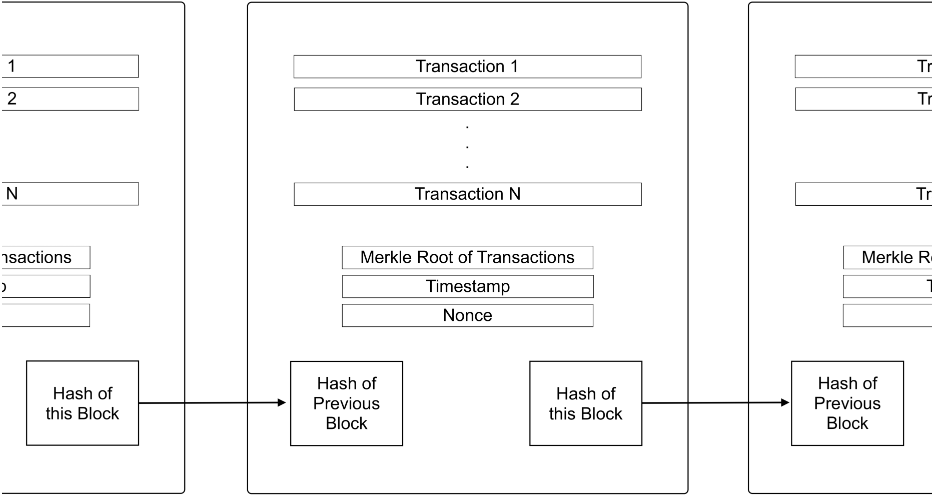
Nakamoto (2008) Blockchain Innovation

- Users submit transactions to a pending transactions list
- Anonymous, decentralized mass of compute power ("miners") competes to validate the next block of transactions
  - Each new block "chains" to previous block
  - Validity: each transaction must (i) be properly signed, (ii) be funded given previous blocks, (iii) not contradict other transactions in this block
- Computational tournament:
  - Brute force search for a "lucky hash" (pseudo-random alphanumeric string) that is a function of: (i) new block, (ii) previous block it chains to
  - Called "proof of work" – hard to find, easy to check

What is Nakamoto Blockchain (2/3)

Nakamoto (2008) Blockchain Innovation

- Users submit transactions to a pending transactions list
- Anonymous, decentralized mass of compute power ("miners") competes to validate the next block of transactions
  - Each new block "chains" to previous block
  - Validity: each transaction must (i) be properly signed, (ii) be funded given previous blocks, (iii) not contradict other transactions in this block
- Computational tournament:
  - Brute force search for a "lucky hash" (pseudo-random alphanumeric string) that is a function of: (i) new block, (ii) previous block it chains to
  - Called "proof of work" – hard to find, easy to check
- Miner who finds a lucky hash reports their new block
  - Other miners check validity (fast), then start working on the next block
  - Winner earns reward paid in bitcoin ("block reward" ≈$100k)



Hash of this Block
Transaction 1
Transaction 2
Transaction N
Merkle Root of Transactions
Timestamp
Nonce
Hash of Previous Block
Hash of this Block
Hash of Previous Block

What is Nakamoto Blockchain (3/3)

- From the Nakamoto (2008) abstract:

  "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 (3/3)

- From the Nakamoto (2008) abstract:

  "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 (3/3)

- From the Nakamoto (2008) abstract:

  "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 (3/3)

- From the Nakamoto (2008) abstract:

  "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)

- Anonymous, decentralized trust.
- 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 more broadly

Critique in 3 Equations

Mining Equilibrium

- P_{block}: reward for winning miner
- c: per-block cost of one unit of computational power
- N: amount of computational power (each $1/N$ chance)
- Free-entry equilibrium:

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

Critique in 3 Equations

Mining Equilibrium

- P_{block}: reward for winning miner
- c: per-block cost of one unit of computational power
- N: amount of computational power (each $1/N$ chance)
- Free-entry equilibrium:

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

Incentive Compatibility (Majority Attack)

- What is cost of majority? $N^{*}c$ per block
- α: expected duration of attack (net of rewards)
- V_{attack}: value of successful attack (discussed below)
- Incentive constraint:

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

Critique in 3 Equations

The Problem

- Together (1) and (2) imply:

Critique in 3 Equations

The Problem

- Together (1) and (2) imply:

$$P_{block} > \frac{V_{attack}}{\alpha} \tag{3}$$

Critique in 3 Equations

The Problem

- Together (1) and (2) imply:

$$P_{block} > \frac{V_{attack}}{\alpha} \quad (3)$$

- 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

- Together (1) and (2) imply:

$$P_{block} > \frac{V_{attack}}{\alpha} \quad (3)$$

- 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

The Problem

- Together (1) and (2) imply:

$$P_{block} > \frac{V_{attack}}{\alpha} \tag{3}$$

- 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
- Economics: *very expensive* form of trust. Memoryless.
  - Usual alternatives: relationships, brands, laws.

Critique in 3 Equations

The Problem

- Together (1) and (2) imply:

$$P_{block} > \frac{V_{attack}}{\alpha} \tag{3}$$

- 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
- Economics: *very expensive* form of trust. Memoryless.
  - Usual alternatives: relationships, brands, laws.
- 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.

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 <u>can</u>
  - 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) subsequently remove the transaction from the blockchain, perhaps after an escrow period

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) subsequently remove the transaction from the blockchain, perhaps after an escrow period

- Under some assumptions, (3) holds on per-transaction basis:

$$p_{transaction} > \frac{\bar{v}_{transaction}}{\alpha}$$

where $\bar{v}_{transaction}$ is a statistic on the largest feasible transactions

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) subsequently remove the transaction from the blockchain, perhaps after an escrow period
- Under some assumptions, (3) holds on per-transaction basis:

$$p_{transaction} > \frac{\bar{v}_{transaction}}{\alpha}$$

  where $\bar{v}_{transaction}$ is a statistic on the largest feasible transactions

- Computational simulations
  - Escrow = 6 blocks → Implicit tax ≈ 30% ($\alpha = 3.35$)
  - Escrow = 1000 blocks → Implicit tax ≈ 2% ($\alpha = 53.5$)
- So if \$1M is easily transacted, tax from \$20k-\$300k *per transaction*

Illustration of Double Spending

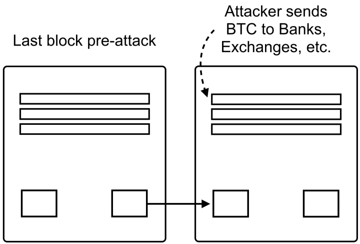


Illustration of Double Spending

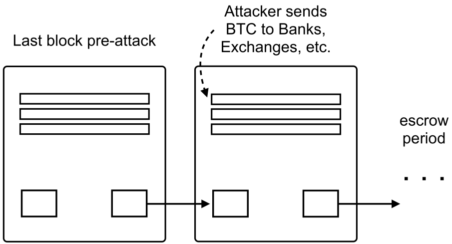


Illustration of Double Spending

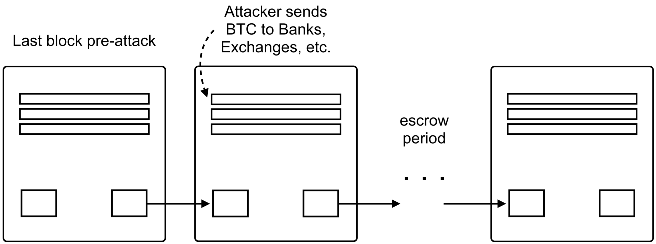


Illustration of Double Spending

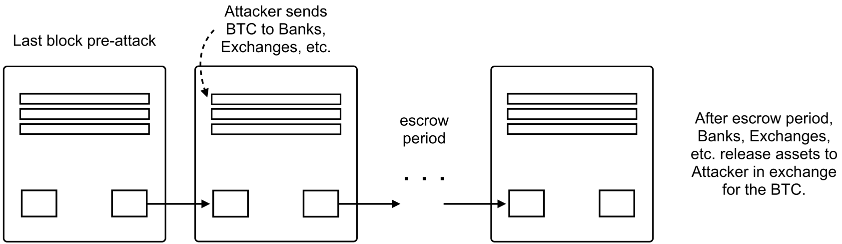


Illustration of Double Spending

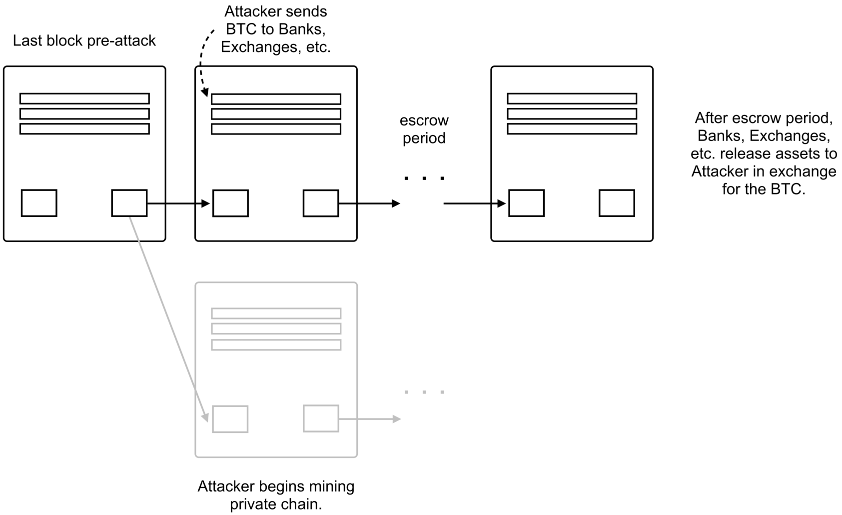


Illustration of Double Spending

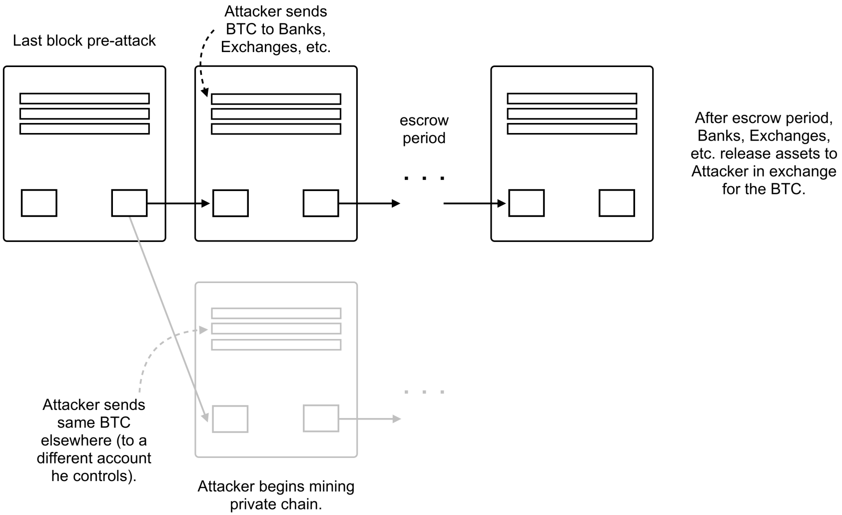


Illustration of Double Spending

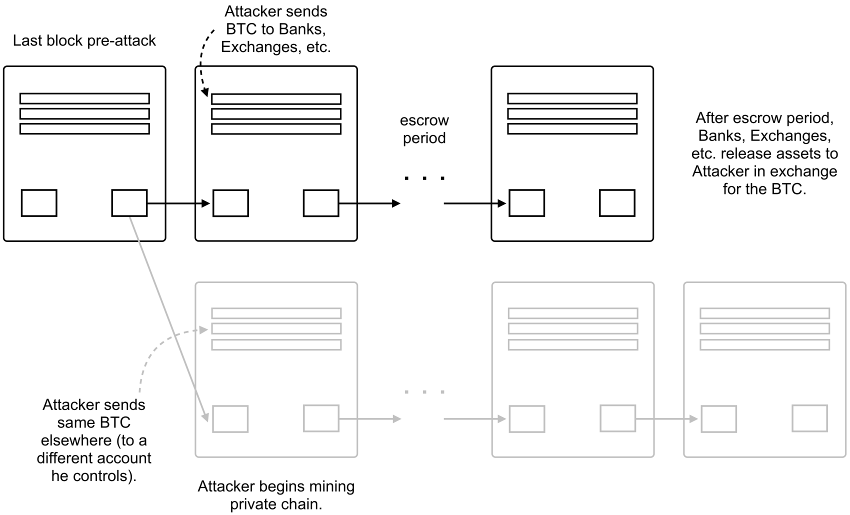
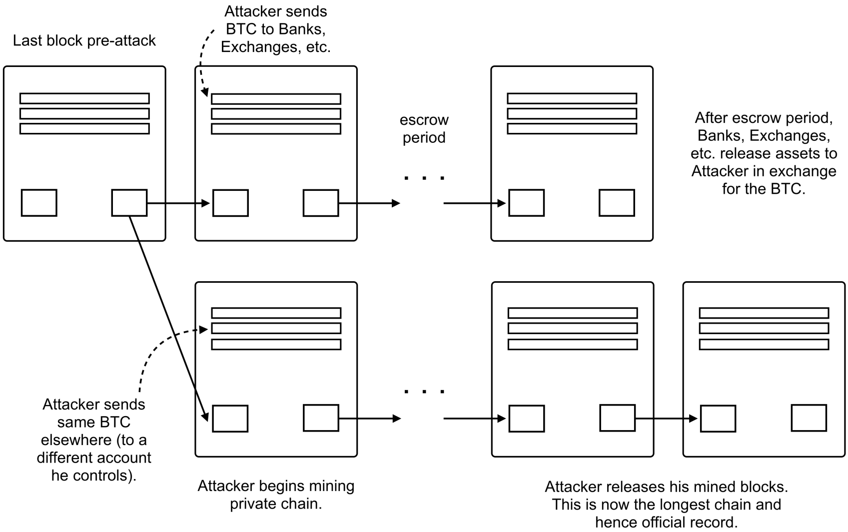


Illustration of Double Spending



Double Spending Attack: Takeaways

$$p_{transaction} > \frac{\bar{v}_{transaction}}{\alpha}$$

- Some takeaways from the double-spending simulations:

Double Spending Attack: Takeaways

$$p_{transaction} > \frac{\bar{v}_{transaction}}{\alpha}$$

- Some takeaways from the double-spending simulations:
- Consistent with early use cases of Bitcoin

Double Spending Attack: Takeaways

$$p_{transaction} > \frac{\bar{v}_{transaction}}{\alpha}$$

- Some takeaways from the double-spending simulations:
- Consistent with early use cases of Bitcoin
- Casts doubt on Bitcoin as "store of value" story (discussed by Cowen, Cochrane, many others)

Double Spending Attack: Takeaways

$$p_{transaction} > \frac{\bar{v}_{transaction}}{\alpha}$$

- Some takeaways from the double-spending simulations:
- Consistent with early use cases of Bitcoin
- Casts doubt on Bitcoin as "store of value" story (discussed by Cowen, Cochrane, many others)
- Casts doubt on Bitcoin as major component of global financial system

Double Spending Attack: Takeaways

$$p_{transaction} > \frac{\bar{v}_{transaction}}{\alpha}$$

- Some takeaways from the double-spending simulations:
- Consistent with early use cases of Bitcoin
- Casts doubt on Bitcoin as "store of value" story (discussed by Cowen, Cochrane, many others)
- Casts doubt on Bitcoin as major component of global financial system
- For the system to be secure for large transactions requires implicit tax rates that render it unusable for small ones

Double Spending Attack: Takeaways

$$p_{transaction} > \frac{\bar{v}_{transaction}}{\alpha}$$

- Some takeaways from the double-spending simulations:
- Consistent with early use cases of Bitcoin
- Casts doubt on Bitcoin as "store of value" story (discussed by Cowen, Cochrane, many others)
- Casts doubt on Bitcoin as major component of global financial system
- 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
  - That is, that α doesn't grow dramatically with e
  - Intuition: attacker earns block rewards while waiting for escrow to clear

Attack II: Sabotage

- If both
  - (i) Mining technology is blockchain-specific / non-repurposible
  - (ii) Attack is a "sabotage", causes large decline in value of Bitcoin
- Then cost of attack is much larger

Attack II: Sabotage

- If <u>both</u>
  - (i) Mining technology is blockchain-specific / non-repurposible
  - (ii) Attack is a "sabotage", causes large decline in value of Bitcoin
- Then cost of attack is much larger
- Consider extreme of 100% collapse
  - Double-spending is pointless
  - Cost is now <u>stock value of the specific capital</u>
- New constraint:

$$N^* C > V_{sabotage} \tag{2'}$$

- \$1.5B-\$2B vs. <\$1M-\$5M.

Attack II: Sabotage

- If both
  - (i) Mining technology is blockchain-specific / non-repurposible
  - (ii) Attack is a "sabotage", causes large decline in value of Bitcoin
- Then cost of attack is much larger
- Consider extreme of 100% collapse
  - Double-spending is pointless
  - Cost is now stock value of the specific capital
- New constraint:

$$N^* C > V_{sabotage} \tag{2'}$$

- \$1.5B-\$2B vs. <\$1M-\$5M.
- However, "pick your poison":

Attack II: Sabotage

- If both
  - (i) Mining technology is blockchain-specific / non-repurposible
  - (ii) Attack is a "sabotage", causes large decline in value of Bitcoin
- Then cost of attack is much larger
- Consider extreme of 100% collapse
  - Double-spending is pointless
  - Cost is now stock value of the specific capital
- New constraint:

$$N^{*}C > V_{sabotage} \tag{2'}$$

- \$1.5B-\$2B vs. <\$1M-\$5M.
- However, "pick your poison":
  - Need to concede possibility of sabotage/collapse

Attack II: Sabotage

- If both
  - (i) Mining technology is blockchain-specific / non-repurposible
  - (ii) Attack is a "sabotage", causes large decline in value of Bitcoin
- Then cost of attack is much larger
- Consider extreme of 100% collapse
  - Double-spending is pointless
  - Cost is now stock value of the specific capital
- New constraint:

$$N^* C > V_{sabotage} \tag{2'}$$

- \$1.5B-\$2B vs. <\$1M-\$5M.
- However, "pick your poison":
  - Need to concede possibility of sabotage/collapse
  - Worry about attacker motivated by sabotage per se

Attack II: Sabotage

- If both
  - (i) Mining technology is blockchain-specific / non-repurposible
  - (ii) Attack is a "sabotage", causes large decline in value of Bitcoin
- Then cost of attack is much larger
- Consider extreme of 100% collapse
  - Double-spending is pointless
  - Cost is now stock value of the specific capital
- New constraint:

$$N^* C > V_{sabotage} \tag{2'}$$

- \$1.5B-\$2B vs. <\$1M-\$5M.
- However, "pick your poison":
  - Need to concede possibility of sabotage/collapse
  - Worry about attacker motivated by sabotage per se
  - Either: high implicit tax rates or risk of collapse

Collapse Scenarios

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

Collapse Scenarios

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

Collapse Scenarios

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

1. Ultra-cheap specialized ASIC chips

Collapse Scenarios

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

1. Ultra-cheap specialized ASIC chips
   - As tech matures: cheap previous-gen versions, or current-gen version becomes cheap enough that electricity becomes the predominant component of cost. Flow cost not stock.

Collapse Scenarios

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

1. Ultra-cheap specialized ASIC chips
   - As tech matures: cheap previous-gen versions, or current-gen version becomes cheap enough that electricity becomes the predominant component of cost. Flow cost not stock.
   - If Bitcoin value falls (for other reasons): glut of chips relative to amount needed for mining equilibrium (1)

Collapse Scenarios

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

1. Ultra-cheap specialized ASIC chips
   - As tech matures: cheap previous-gen versions, or current-gen version becomes cheap enough that electricity becomes the predominant component of cost. Flow cost not stock.
   - If Bitcoin value falls (for other reasons): glut of chips relative to amount needed for mining equilibrium (1)
2. Efficient-enough repurposable chips
   - If blockchain grows in importance and repurposable chips get better at hashing. Then flow cost not stock.

Collapse Scenarios

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

1. Ultra-cheap specialized ASIC chips
   - As tech matures: cheap previous-gen versions, or current-gen version becomes cheap enough that electricity becomes the predominant component of cost. Flow cost not stock.
   - If Bitcoin value falls (for other reasons): glut of chips relative to amount needed for mining equilibrium (1)
2. Efficient-enough repurposable chips
   - If blockchain grows in importance and repurposable chips get better at hashing. Then flow cost not stock.
3. Economic sabotage becomes sufficiently tempting
   - Futures markets grow
   - Bitcoin grows in economic importance

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
Monacoin	Lyra2rev2	5/13/2018	$90,000
Bitcoin Gold	Equihash	5/16/2018	$18,000,000
Litecoin Cash	SHA-256	5/30/2018	Unknown
Zencash	Equihash	6/2/2018	$700,000
Vertcoin	Lyra2rev2	10/12/2018	$100,000
Ethereum Classic	Ethash	1/5/2019	$1,100,000

Sources: Coindesk, Bitcoinist, CCN, and Cointelegraph. 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
Monacoin	Lyra2rev2	5/13/2018	$90,000
Bitcoin Gold	**Equihash**	**5/16/2018**	**$18,000,000**
Litecoin Cash	SHA-256	5/30/2018	Unknown
Zencash	Equihash	6/2/2018	$700,000
Vertcoin	Lyra2rev2	10/12/2018	$100,000
Ethereum Classic	Ethash	1/5/2019	$1,100,000

Sources: Coindesk, Bitcoinist, CCN, and Cointelegraph. 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
  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
  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
  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.

Conclusion: Remark

- Emphasize: model consistent with earliest uses of Bitcoin and blockchain
- 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
- 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: <u>not</u> skeptical of use of distributed databases more broadly

Conclusion: Remark

- Emphasize: model consistent with earliest uses of Bitcoin and blockchain
- 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