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

Eric Budish
University of Chicago, Booth School of Business

October 12th, 2022
Stanford Theory Seminar

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, anonymous, freely-entering and -exiting mass of computing power around the world to pay attention to and collectively maintain a common data set
- Enabling trust in this data set
  - (CS terminology for the invention: "permissionless consensus")

- 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
- Yet, economic usefulness remains an open question
- To date, majority of volume appears speculative, with other widely-documented use case being black market (Makarov and Schoar, 2021; Foley et al., 2019; Yellen, 2021; Gensler, 2021)
  - Ironically, most of the speculative volume is through cryptocurrency exchanges — which are trusted financial intermediaries

- 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 ..."

This Paper's Argument

- This paper argues that Bitcoin and the Nakamoto blockchain — while undeniably ingenious — have serious economic limitations
- Analysis ultimately suggests skepticism that Bitcoin and Nakamoto's anonymous, decentralized trust will play a major role in the global economy and financial system

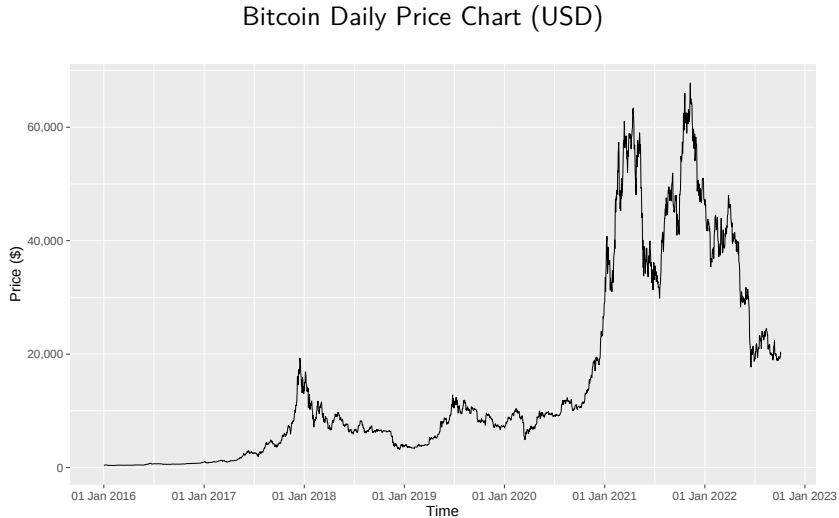
This Paper's Argument

- Core of the argument is just 3 equations.
- Equation (1): zero-profits condition.
  - The amount of computing power devoted to maintaining the trust reflects the compensation paid to this computing power (called "miners").
- Equation (2): incentive compatibility condition.
  - How much trust does a given level of computing power produce?
  - Vulnerability: "majority attack".
  - IC: costs of attack must exceed the benefits.
- 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!
  - Especially as stakes grow! Scales linearly.
- Intuition: blockchain trust is "memoryless"
- Under idealized attack circumstances, get an even stronger result:
  - "Zero net attack cost theorem"

This Paper's Argument

- So ... why hasn't Bitcoin already been attacked?
- A way out of the "extremely expensive" argument:
  - (i) mining technology is specialized/non-repurposable, and
  - (ii) majority attack causes collapse
- Why? Makes attack much more expensive.
  - Attacker pays not just the "flow" cost of attack, but the "stock" value of the now-worthless specialized mining computers.
  - 3-4 orders of magnitude difference in costs.
- This is good news about security costs, but vulnerability to collapse is itself a serious problem.
  - Especially if thinking about cryptocurrencies playing a meaningful role in global financial system.
  - "Pick your poison"
- Analysis points to specific collapse scenarios.

A Disclaimer ...



Source: www.wsj.com

A Disclaimer ...

- I have no explanation for Bitcoin's asset value (currrently about $400bn, 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, 2021)
  - 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
  - Free Entry Condition (Miners)
  - Incentive Compatibility Condition (Majority Attack)
  - Economic Limits: Flow vs. Stock Problem. Zero Net Attack Cost Theorem.
- Analysis of Double Spending Attacks
- A Way Out: Specialized Capital + Risk of Collapse
  - A Softer Constraint: Stock vs. Stock
  - Collapse Scenarios
- Open Questions

Overview of the Talk

- **Overview: What is Nakamoto Blockchain**
- Nakamoto Blockchain: A Critique in 3 Equations
  - Free Entry Condition (Miners)
  - Incentive Compatibility Condition (Majority Attack)
  - Economic Limits: Flow vs. Stock Problem. Zero Net Attack Cost Theorem.
- Analysis of Double Spending Attacks
- A Way Out: Specialized Capital + Risk of Collapse
  - A Softer Constraint: Stock vs. Stock
  - Collapse Scenarios
- Open Questions

What is Nakamoto Blockchain (1/4)

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

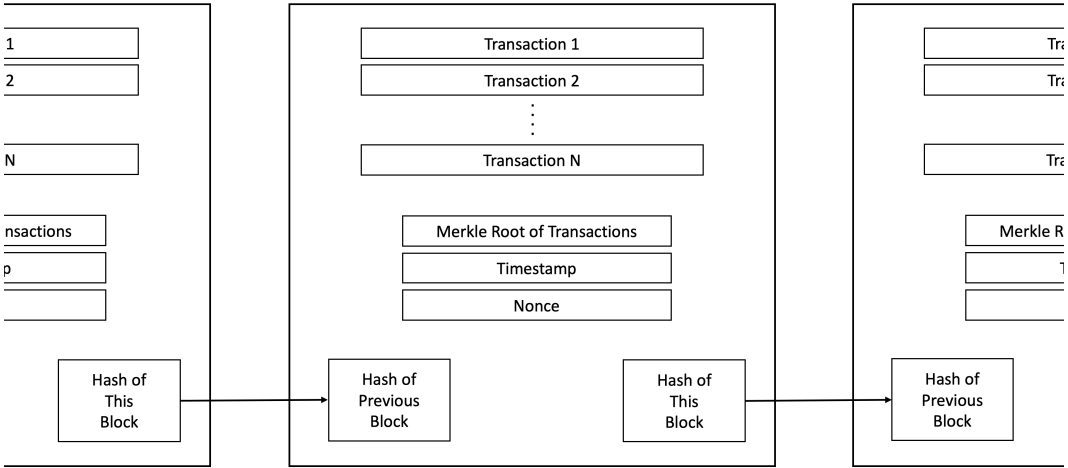
Sender	Receiver	Amount	Signature
Alice	Bob	$10	*Alice*

- **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)

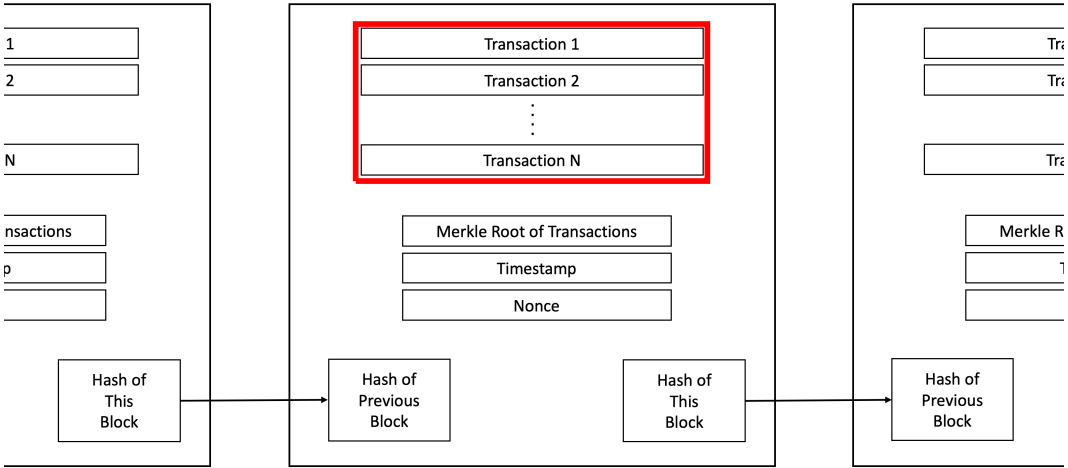
Nakamoto (2008) Blockchain Innovation

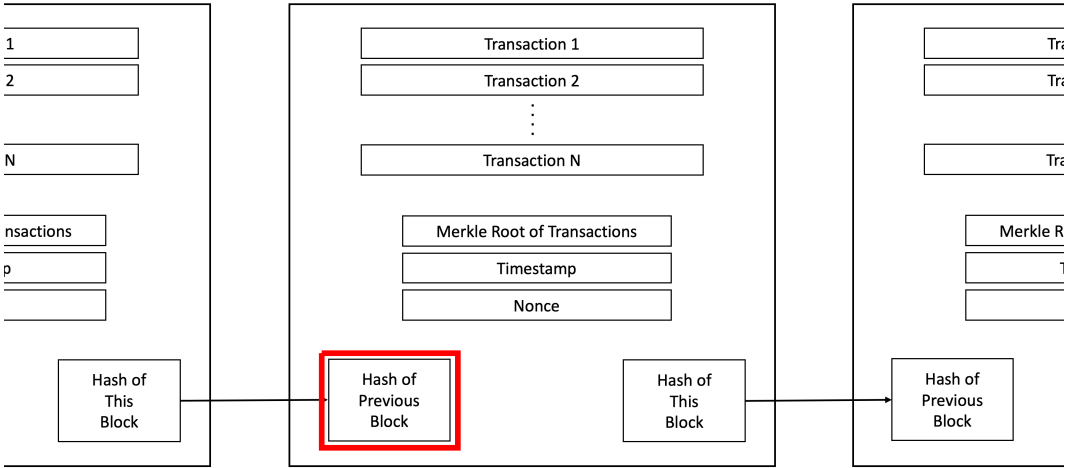
- **I: Pending Transactions List**
  - Users submit transactions to a pending transactions list, called mempool
  - Like a google spreadsheet — not considered official yet
- **II: Valid Blocks**
  - Any computer around the world can compete for the right to add transactions from the mempool to a data structure called the blockchain. (Will describe competition next)
  - Each new block of transactions "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 transaction 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.

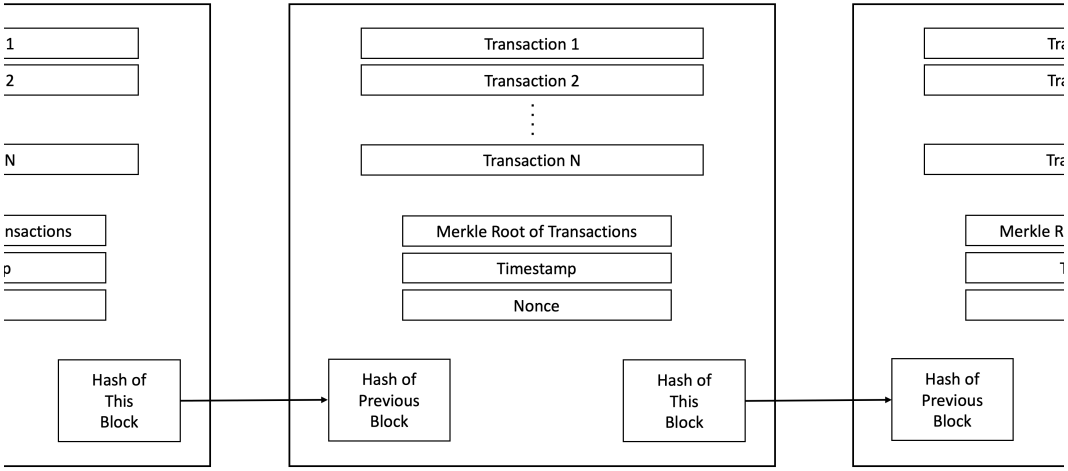
What is Nakamoto Blockchain (3/4)

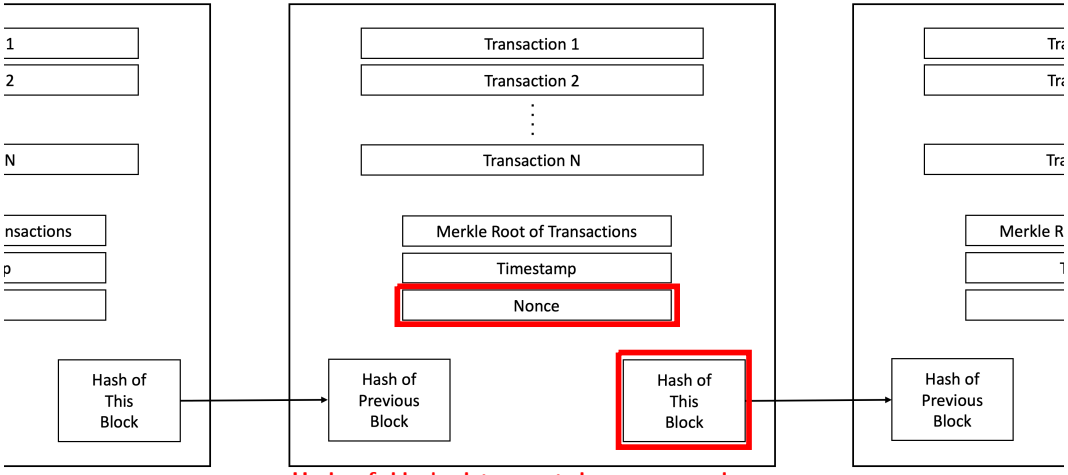
- **III: Bitcoin "Mining" Computational Tournament**
  - Boils down to a massive brute-force search for a lucky random alphanumeric string
  - Free entry, free exit, all anonymous. Anyone can play at any time.
  - "Miner" chooses a valid block of transactions from the mempool
  - Then searches for an 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
  - Example: block 729,999 has the hash

$$000000000000000000008b6f6fb83f8d74512ef1e0af29e642dd20daddd7d318f$$

  - Called "proof of work" – hard to find, easy to check (pseudorandom, non-invertible)
  - Current hash rate: about 200 million TH/s (2 x 10^20)

- 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"). Currently 6.25 new Bitcoins plus small transaction fees.
  - 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 block data must have a very large number of leading zeros.

Example from Block $729,999$**:**

- **Nonce:** 3477019455
- **Hash:** $0000000000000000008b6f6fb83f8d745\ldots$

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: *the official consensus record of transactions is the longest chain, as measured by the amount of computational work*
  - Intuition #1: as long as a majority of mining power is "honest" and follows the longest chain, then the longest chain will stay longest with probability one
    - Computing power like "votes" -> enables decentralized adjudication of which is the official chain if there are multiple
    - What makes the Bitcoin blockchain real and the "Budish blockchain" (run from my laptop) an imposter? Answer: the work.
  - Intuition #2: need some decentralized way to coordinate miner's efforts
    - Honest mining is a Nash equilibrium of Nakamoto longest-chain if all miners are "small" (Kroll et al. (2013), Carlsten et al. (2016), Biais et al. (2019))
  - But note: vulerable to attack by a 51% majority. Can outpace honest miners with probability one.
    - (Not surprising that it is vulnerable. Decentralized consensus that pre-dates Nakamoto, based on Byzantine Fault Tolerance, vulnerable to $\frac{1}{3}$ attack)

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 on-going 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 I: "Permissioned Blockchains"

- 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

Clarification II: "Smart Contracts"

- Notice that Nakamoto's novel form of trust isn't specific to currency transactions
- Can replace "Alice sends Bob 10 BTC, signed by Alice" with any executable computer instruction signed by Alice.
- This idea is often called "smart contracts". Analysis framework of this paper applies analogously
  - Though attack possibilities will differ (e.g., no such thing as double spending per se if the code is not executing currency transactions).

Clarification III: Proof of Stake

- "Proof of Stake" as opposed to Proof of Work
- Roughly: instead of voting for the correct chain with computational work, vote with stake in the cryptocurrency
  - Ethereum recently switched from proof-of-work to proof-of-stake
  - Several other blockchains use proof-of-stake
- Usual motiviation: reduce mining expense and environmental harm ("Ethereum reduces its energy use by 99.95%")
- Environmental issue is orthogonal to the concerns raised in this paper
- What's interesting re this paper's argument is that stakes have *memory*. This opens up new possibilities for making attacks more expensive
  - Will return to this at the end
  - So far, no PoS that makes "all" attacks more expensive. (Ex: Ethereum PoS makes double-spending attacks much more expensive, but is vulnerable to "liveness attacks" which are cheap. Where "expensive" = stock, "cheap" = flow.).

Overview of the Talk

- Overview: What is Nakamoto Blockchain
- **Nakamoto Blockchain: A Critique in 3 Equations**
  - **Free Entry Condition (Miners)**
  - **Incentive Compatibility Condition (Majority Attack)**
  - **Economic Limits: Flow vs. Stock Problem. Zero Net Attack Cost Theorem.**
- Analysis of Double Spending Attacks
- A Way Out: Specialized Capital + Risk of Collapse
  - A Softer Constraint: Stock vs. Stock
  - Collapse Scenarios
- Open Questions

Zero-Profit Condition (Blockchain Miners)

- Conceptual question: how much computational power will maintain Nakamoto's anonymous, decentralized trust, if we restrict all to behave honestly?

- Treat time as continuous
- N: amount of computational power
  - Large finite number of honest miners
  - Follow longest chain protocol automatically
  - Player i chooses qty of computing power x_i. Define $N = \sum_i x_i$.
  - Eqm concept will be zero-profit. Captures permissionless, free entry/exit.

- p_{block}: compensation per block paid to the miner that wins the computational tournament
  - Assume exogenous. Will derive constraints below.
  - Proportional rule: player i wins a given block with prob. $\frac{x_i}{N}$

- c: cost per unit time to run one unit of computing power
  - Includes rental cost of capital and variable costs ($c = rC + \eta$)
  - Can generalize to have an upward sloping supply curve

Zero-Profit Condition (Blockchain Miners)

- D: block difficulty level. Defined as how many units of compute-time are needed in expectation to solve one block (assume Poisson arrivals)
- Honest miner profits: if N units of computing power, D difficulty
  - Some miner solves a block every $\frac{D}{N}$ time in expectation.
  - Profits per unit of compute per unit time are thus

$$\frac{1}{N}\frac{D}{N}p_{block} - c$$

- Definition. A zero-profit honest mining equilibrium consists of quantities $\{x_i^*\}_{i \in I}$ and a difficulty level D^* such that miners (i) solve one block per unit time (as a normalization), and (ii) earn zero economic profits in expectation.
- Result: Let $N^* = \sum_i x_i^*$. In any zero-profit honest mining equilibrium, $D^* = N^*$ and

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

- Note: (1) widely known (many papers, Bitcoin Wiki).
- Note: if use Nash eqm for entry, still restrict to honest play, then $N^* c < p_{block}$

Incentive Compatibility (Majority Attack)

- Conceptual question: how much security is generated by the amount of honest mining in (1)?
- Vulnerability: an attacker with $> 50\%$ of total computational power can double-spend with probability one.
- Attack costs
  - Consider an additional player, the attacker, not restricted to honest play.
  - Can attack by choosing AN^* units of computing power, $A > 1$, for an $\frac{A}{A+1}$ majority
  - Cost per unit time: AN^*c
  - Expected duration of attack: $t(A)$. Will derive closed form in next section under assumptions.
  - Call $AN^*c \cdot t(A)$ the gross cost of attack.
- Attacker can minimize $A \cdot t(A)$: call this $A^* \cdot t(A^*)$
- Let V_{attack} denote the value of an attack
  - For now, abstract. Will derive a constraint in relation to p_{block}
  - Should have in mind that the value of attack will grow as Bitcoin's importance / usefulness grow.

Incentive Compatibility (Majority Attack)

- Definition. The blockchain is incentive compatible against an outsider attack, on a gross-cost basis, if the gross cost of attack exceeds the benefits of attack:

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

- Remarks
- Inside vs. Outside Attacker
  - (2) is the IC for an outside attacker.
  - An attack could also come from the inside — part of the current honest mining. Cheaper: as little as $\frac{N^* c}{2}$ per unit time
  - Outside attacker seems more attractive as a conceptual approach. Treats the honest miners as "small" which is the Nakamoto ideal. Honest as an atomless continuum that behaves automatically, fluctuates in size with p.
  - Inside attacker might be more realistic in practice. Cheaper, already have the equipment, and miners are concentrated (Makarov and Schoar; Cong, He and Li)
- Gross vs. Net Cost
  - (2) is a gross cost. In Bitcoin, attacker would earn block rewards for the blocks in their new chain, so Net $<$ Gross. Will come back to this

Critique in 3 Equations

The Problem

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

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

- Proposition. *The zero-profit condition (1) and gross incentive-compatibility condition (2) together imply the equilibrium constraint:*

$$p_{block} > \frac{V_{attack}}{A^* \cdot t(A^*)} \tag{3}$$

- *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

$$p_{block} > \frac{V_{attack}}{A^* \cdot t(A^*)}$$

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

Net Cost of Attack and a "Zero" Theorem

- 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 performs $At \cdot N^*$ compute-units of work.
  - If difficulty stays constant at $D^{'} = D^* = N^*$, earns At block rewards in expectation
- 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 reduces value of block rewards, value of Bitcoins kept in double-spend attack. (Assume for now capital is repurposable and retains its value.)
  - Let $\Delta_{attack} \geq 0$ parameterize decline.
    - Reduces block rewards by $\Delta_{attack} At \cdot N^* c$
    - Reduces benefit of attack by $\Delta_{attack} V_{attack}$

Net Cost of Attack and a "Zero" Theorem

- Theorem: *if the attacker's cost is the same as honest miners ($\kappa = 0$), the attack concludes before difficulty adjusts ($D^{'} = N^{*}$), and the attack does not cause the value of Bitcoin to fall ($\Delta_{attack} = 0$), then the net cost of attack is zero.*
- Proof:
  - Computational cost of attack: $(1 + \kappa)At \cdot N^{*}c$
  - Net value of block rewards: $At \cdot \frac{N^{*}}{D^{'}} p_{block}(1 - \Delta_{attack})$
  - If $\kappa = \Delta_{attack} = 0$, $D^{'} = N^{*}$, and using equation (1), then computational costs less net value of block rewards is

$$At \cdot N^{*}c - At \cdot N^{*}c = 0$$

- Intuition: attacker is fully compensated for their computational costs for same reason as honest miners are fully compensated for their costs under honest play.
- Implication: Bitcoin's security relies on either attacker cost frictions or the presumption that attacks would cause a large decline in the value of Bitcoin.
- (To be clear: zero frictions and zero decline seem unrealistic, but are useful as a benchmark case.)

A One-Shot Game Version of (1)-(3)

- Some of the complexity in analysis relates to timing issues and/or conventions specific to Bitcoin
  - Costs are per unit time
  - Payments are per block – stochastic arrivals
  - Attack duration is stochastic
  - Difficulty adjustment
- Consider instead the following simplified one-shot game
- I "nodes". (Work, stake, etc.)
- Each node i chooses:
  - Quantity x_i
  - Posture $a_i \in \{Honest, Attack\}$
- Cost is c per unit. Define $N = \sum x_i$.
- Payoffs:
  - If there is a player i with $x_i > \frac{N}{2}$ and $a_i = Attack$: player i gets V_{attack}
  - Else: each player i gets $\frac{x_i}{N} p$

A One-Shot Game Version of (1)-(3)

- Question: under what conditions is there a Nash equilibrium in which all players i choose $a_i = \textit{Honest}$ (and some x_i^* consistent with NE)
  - Lemma. If there is an honest equilibrium, then $N^* c \leq p$. (1)
  - Theorem. A necessary condition for no player to have a profitable attack is $p \geq \frac{V_{attack}}{1+\frac{1}{I}}$ (3)
- Proof of Theorem.
  - Honest play payoff for i: $\frac{x_i^*}{N^*} p - x_i^* c$
  - Attack payoff for i: $V_{attack} - N_{j \neq i}^* c$ (where $N_{j \neq i}^* = \sum_{j \neq i} x_i^*$)
  - Need: $V_{attack} - N_{j \neq i}^* c \leq \frac{x_i^*}{N^*} p - x_i^* c$. (If $x_i^* = 0$, this is $N^* c \geq V_{attack}$, which corresponds to (2))
  - Rearrange: $V_{attack} \leq N_{j \neq i}^* c - x_i^* c + \frac{x_i^*}{N^*} p$
  - Using Lemma: $V_{attack} \leq p + \frac{x_i^*}{N^*} p$
  - Using smallest x_i^*: $V_{attack} \leq p(1 + \frac{1}{I})$. QED.
- As I goes to infinity, condition is $p \geq V_{attack}$
- Interpretation: p, c, now both represent a unit of time commensurate with duration of attack. (Analog of $A^* \cdot t(A^*)$ in (3))

Overview of the Talk

- Overview: What is Nakamoto Blockchain
- Nakamoto Blockchain: A Critique in 3 Equations
  - Free Entry Condition (Miners)
  - Incentive Compatibility Condition (Majority Attack)
  - Economic Limits: Flow vs. Stock Problem. Zero Net Attack Cost Theorem.
- **Analysis of Double Spending Attacks**
- A Way Out: Specialized Capital + Risk of Collapse
  - A Softer Constraint: Stock vs. Stock
  - Collapse Scenarios
- Open Questions

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 their 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

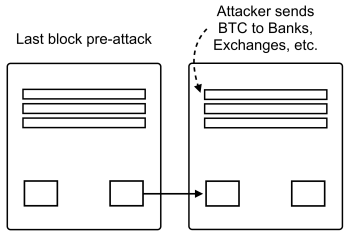


Illustration of Double Spending

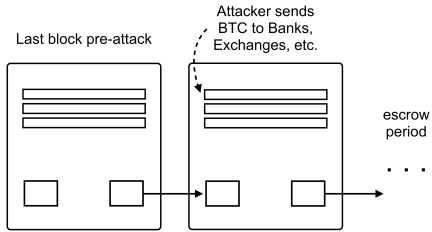


Illustration of Double Spending

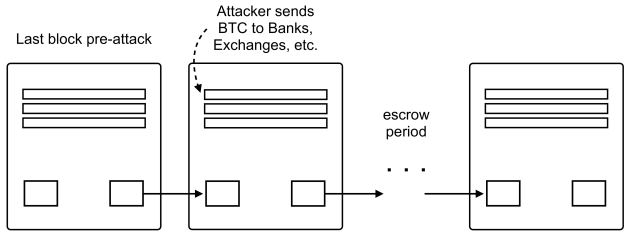


Illustration of Double Spending

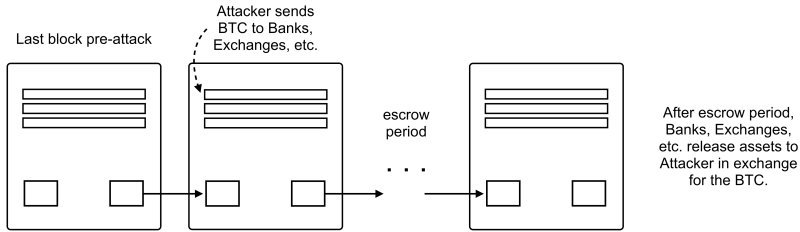


Illustration of Double Spending

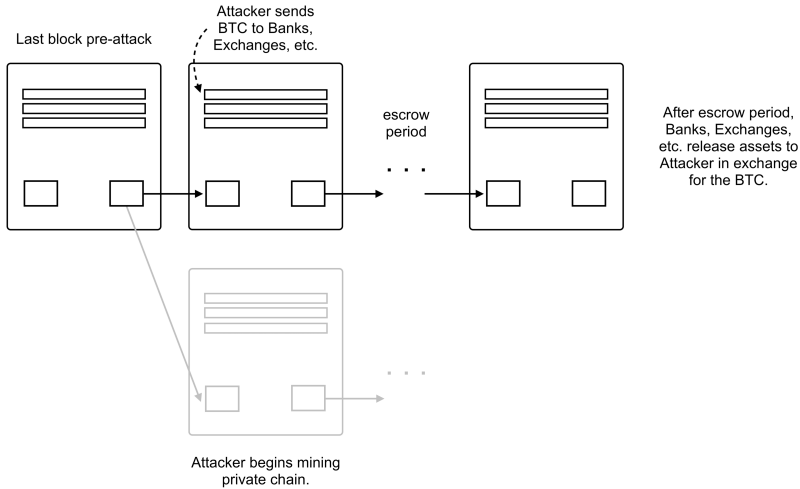


Illustration of Double Spending

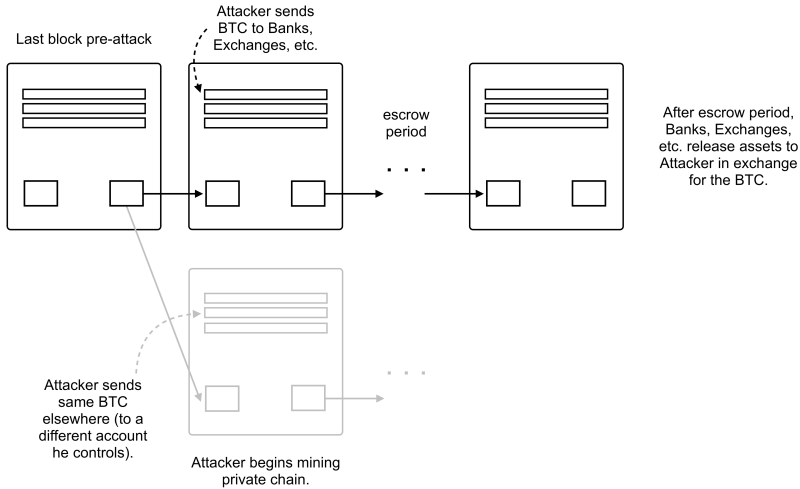


Illustration of Double Spending

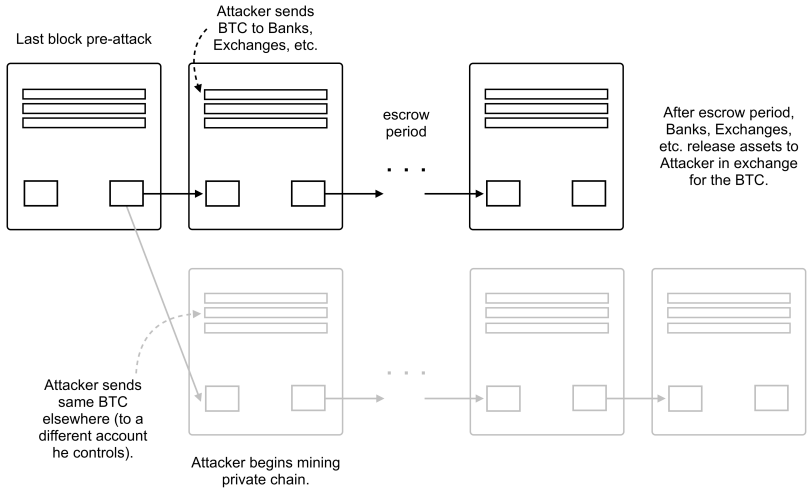
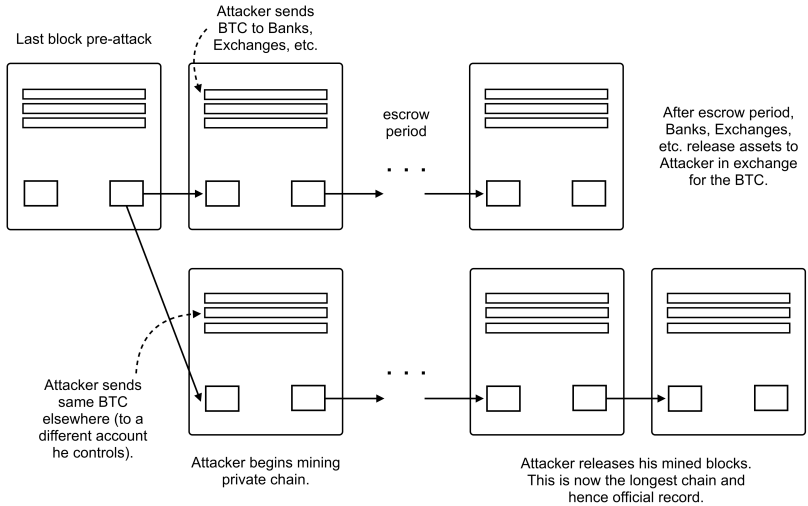


Illustration of Double Spending



Double Spending: Analysis Framework

- Equation (3) tells us that the possibility of a double-spending attack places an economic limit on Nakamoto trust:

$$p_{block} > \frac{V_{attack}}{A^* \cdot t(A^*)}$$

- Benefits of attack: V_{attack}
  - 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)
  - Interpretation: V_{attack} represents the amount of transaction volume that *honest users* of Bitcoin can conduct in a short period of time
  - I consider a range from $1000 (pizza) to $100bn (global finance)
- Duration of attack: $A^* \cdot t(A^*)$
  - Can compute explicitly
- Then ask: how big need p_{block} be for a given desired amount to secure, V_{attack}

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
- Proposition. 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

- Consider a range of cases for At informed by the computations
- **Base case,** $At = 16$. Corresponds to gross costs under current escrow period $e = 6$ and attacker majority $A = 1.25$ (55%).
  - Net costs if $\kappa = 1$ (frictions cancel block rewards) and $\Delta_{attack} = 0$.
- **Expensive attack case**, $At = 150$. Corresponds to one full day of block-compute-costs.
  - Appropriate if escrows longer for higher-value transactions.
  - Or, base case with higher attack frictions.
- **Very expensive attack case** $At = 1000$. 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 $\frac{p_{block}}{V_{attack}} \geq \frac{1}{At}$
- 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 $\frac{p_{block}}{V_{attack}} \geq \frac{1}{At}$
- 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 $\frac{p_{block}}{V_{attack}} \geq \frac{1}{At}$
- 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 $\frac{p_{block}}{V_{attack}} \geq \frac{1}{At}$
- 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. Securing Against an Attack: Sensitivity Analysis

Attack Scenarios	Per-Block	Per-Day	Per-Year	Per-Transaction
Base Case	6.25 %	900 %	328,500 %	0.003 %
Expensive	0.67 %	96 %	35,040 %	0.0003 %
Very Expensive	0.10 %	14 %	5,256 %	0.00005 %

- 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. $1B attack) requires fees that are prohibitive for small transactions
  - Even at a 1 week attack duration ($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. Securing Against an Attack: Sensitivity Analysis

Attack Scenarios	Per-Block	Per-Day	Per-Year	Per-Transaction
Base Case	6.25 %	900 %	328,500 %	0.003 %
Expensive	0.67 %	96 %	35,040 %	0.0003 %
Very Expensive	0.10 %	14 %	5,256 %	0.00005 %

- 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. $1B attack) requires fees that are prohibitive for small transactions
  - Even at a 1 week attack duration ($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. Securing Against an Attack: Sensitivity Analysis

Attack Scenarios	Per-Block	Per-Day	Per-Year	Per-Transaction
Base Case	6.25 %	900 %	328,500 %	0.003 %
Expensive	0.67 %	96 %	35,040 %	0.0003 %
Very Expensive	0.10 %	14 %	5,256 %	0.00005 %

- 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. $1B attack) requires fees that are prohibitive for small transactions
  - Even at a 1 week attack duration ($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

$$p_{block} > \frac{V_{attack}}{At}$$

- 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 a long-enough escrow period isn't enough
  - Missed that need to worry about larger and larger attacks if Bitcoin gets more economically useful

Overview of the Talk

- Overview: What is Nakamoto Blockchain
- Nakamoto Blockchain: A Critique in 3 Equations
  - Free Entry Condition (Miners)
  - Incentive Compatibility Condition (Majority Attack)
  - Economic Limits: Flow vs. Stock Problem. Zero Net Attack Cost Theorem.
- **A Way Out: Specialized Capital + Risk of Collapse**
  - **A Softer Constraint: Stock vs. Stock**
  - **Collapse Scenarios**
- Open Questions

Attack II: Sabotage

- Obvious response: double spending attack would be "noticed"
- Cause decline in value of Bitcoin, which attacker will be left with after a double spend (V_{attack} worth)
- Bitcoin Wiki classifies majority attack "Probably Not a Problem" for this reason
- As above, suppose attack causes Bitcoin value to decline by proportion Δ_{attack}. Attacker cost frictions κ. Equation (3) becomes:

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

- Proposition. *For any potential value of a double-spending attack V_{attack}, and any level of block reward p_{block}, the Bitcoin blockchain is secure against the double-spending attack if Δ_{attack} is sufficiently large.*
- This may sound reassuring about security ...
  - But the argument concedes that an attack would cause collapse of the trust
  - Raises worry about attacker motivated by collapse per se
  - **Pick your poison: high implicit tax rates or risk of collapse**

Attack II: Sabotage

- How big is V_{attack} from a sabotage?
- Hard to say, but seems likely to already be large relative to the Base, Expensive, and maybe even Very Expensive gross costs of attack ($4M - $250M at recent values)
- 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
- Bitcoin market capitalization: as high as $1 trillion (Peter Thiel prediction: $100 trillion)
- Vitalik Buterin: "if blockchains do become successful enough, and they survive long enough, they have a good enough track record of actually being the base layer for many kinds of interactions, and we fast-forward a couple of decades into a future where it's **just considered normal for there to be trillion dollar assets that are managed on Ethereum** ..." (Ezra Klein podcast, Sept 30, 2022)

Sabotage and Blockchain-Specific Capital

- Why would a sabotage attack cost a stock, not a flow?
- 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
- If capital is specialized, and attack causes collapse, then the attacker cost model needs to be modified
  - In addition to charging attacker a flow cost that is $O(N^* c)$, where $c = rC + \eta$
  - Also need to charge attacker the value of the now-worthless specialized capital: $O(N^* C)$

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 May 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 $c = rC + \eta$. Honest mining equilibrium (1) can be written as:

$$N^* c = N^*(rC + \eta) = p_{block}. \tag{1}$$

- Outside attacker needs $N^* C$ of capital. Assume attack causes total collapse of the trust. IC constraint to secure against outsider sabotage is approximated by

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

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

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

- Hence we can derive a modified version of (3):

$$p_{block} > \frac{r}{\mu} V_{attack} \tag{3'}$$

Cost to Secure Against Sabotage, Derivation

- MUCH more secure than before, because of r (interest rate per block!). So relative to original, improve security by several orders of magnitude.
- Sense of magnitudes
  - The change in the IC constraint is a factor of $At\frac{r}{\mu}$
  - If we use base case of $At = 16$, use $r = 50\%$ annually which is $\sim 0.001\%$ per block, and $\mu = 0.4$, we have $At\frac{r}{\mu} = 0.0004$. A 2500x reduction in the rewards necessary for security.
  - (N.B. these values of r and μ, with 2022 avg. values of p_{block}, imply $N^*C = \$12B$ which roughly matches observed prices.)
- Annual cost to secure $1bn:
  - Original model without collapse: $3.3 trillion
  - Sabotage model with collapse: $1.25 billion ($2.5 bn for insider sabotage)
- Current capital stock and miner payments suggests Bitcoin is secure up to sabotages worth roughly $10bn for an outsider, $5bn for an insider

Collapse Scenarios

- So we have a candidate answer to the Chicago Lunch Table question: Bitcoin hasn't been attacked yet because of (i) specialized equipment, and (ii) attackers would lose the stock value of their specialized equipment in an attack, becaus an attack will cause the system to collapse. And this stock cost of attack is larger than the current attack possibilities.
- Suppose this is right. That is:
  - Bitcoin blockchain *does not* satisfy (2): $A^* N^* c \cdot t(A^*) > V_{attack}$
  - Bitcoin blockchain *does* satisfy (2'): $N^* C > V_{attack}$
  - Attack would cause collapse, hence (2') not (2) is operative
- Question: what changes to the economic environment could cause the binding constraint to change from (2') to (2)? Or cause (2') no longer to hold?

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-compute electricity cost is $\eta' > c$. So in honest mining equilibrium, old chips are not economical to use even if the chips themselves are free.
- Observation: If there are $\geq N^*$ compute units of old chips, and these chips are approximately free, then attacker can attack at flow cost of $N^*\eta'$.
- 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, so this could change.
- More generally, if security depends on specialized chips, then Bitcoin is vulnerable to changes in the chip market.

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 some capital will be "mothballed". Not worth the variable costs even if treat capital as free.
- 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
- Example: suppose N^* reflects $p_{block} = \$250k$, early 2022 avg.
  - By 2032, at current prices, $p_{block} \approx \$20k$. 92% decline. Even if capital is free, only $\frac{1-.92}{1-\mu} = 13\%$ will be used. 87% will be mothballed.

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 logical possibility: Bitcoin grows in economic importance enough to tempt a saboteur despite the cost
  - That is, (2') fails to hold: $V_{attack} > N^* C$.
- 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
Bitcoin SV	SHA-256	8/3/2021	Unknown
Verge	Scrypt, X17, Lyra2rev2, Myr-groestl, Blake2s	2/15/2021	Unknown
		4/4/2018	$2.8 million
Grin	Cuckoo Cycle	11/8/2020	Unknown
Ethereum Classic	Ethash	8/29/2020	Unknown
		8/6/2020	$1.7 million
		7/29/2020	$5.6 million
		1/5/2019	$1.1 million
Bitcoin Gold	Equihash	1/23/2020	$100 thousand
		5/16/2018	$18 million
Firo	MTP	1/19/2019	$4 million
Vertcoin	Lyra2rev2	10/12/2018	$100 thousand
Zencash	Equihash	6/2/2018	$700 thousand
Litecoin Cash	SHA-256	5/30/2018	Unknown
Monacoin	Lyra2rev2	5/13/2018	$100 thousand

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
Bitcoin SV	SHA-256	8/3/2021	Unknown
Verge	Scrypt, X17, Lyra2rev2, Myr-groestl, Blake2s	2/15/2021	Unknown
		4/4/2018	$2.8 million
Grin	Cuckoo Cycle	11/8/2020	Unknown
Ethereum Classic	**Ethash**	**8/29/2020**	**Unknown**
		8/6/2020	**$1.7 million**
		7/29/2020	**$5.6 million**
		1/5/2019	**$1.1 million**
Bitcoin Gold	**Equihash**	1/23/2020	$100 thousand
		5/16/2018	**$18 million**
Firo	MTP	1/19/2019	$4 million
Vertcoin	Lyra2rev2	10/12/2018	$100 thousand
Zencash	Equihash	6/2/2018	$700 thousand
Litecoin Cash	SHA-256	5/30/2018	Unknown
Monacoin	Lyra2rev2	5/13/2018	$100 thousand

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 Crypto Thefts

Name	Type of Business	Date of Attack	Amount Stolen
BNB Chain	DeFi Bridge	October 2022	$568 million
Wintermute	DeFi Market Maker	September 2022	$160 million
Nomad	DeFi Bridge	August 2022	$200 million
Beanstalk Farms	DeFi Stablecoin	April 2022	$182 million
Ronin Network	DeFi Bridge	March 2022	$625 million
Wormhole	DeFi Bridge	February 2022	$320 million
BitMart	Centralized Exchange	December 2021	$150 million
C.r.e.a.m. Finance	DeFi Lending Protocol	October 2021	$130 million
PolyNetwork	DeFi Bridge	August 2021	$600 million
KuCoin	Centralized Exchange	September 2020	$281 million
BitGrail	Centralized Exchange	February 2018	$170 million
Coincheck	Centralized Exchange	January 2018	$530 million
Mt. Gox	Centralized Exchange	February 2014	$480 million

Sources: Bloomberg, WSJ, Elliptic Inc. Amounts calculated based on fund values at the time of theft.

Examples of Crypto Thefts

Name	Type of Business	Date of Attack	Amount Stolen
BNB Chain	DeFi Bridge	October 2022	$568 million
Wintermute	DeFi Market Maker	September 2022	$160 million
Nomad	DeFi Bridge	August 2022	$200 million
Beanstalk Farms	**DeFi Stablecoin**	**April 2022**	**$182 million**
Ronin Network	DeFi Bridge	March 2022	$625 million
Wormhole	DeFi Bridge	February 2022	$320 million
BitMart	Centralized Exchange	December 2021	$150 million
C.r.e.a.m. Finance	DeFi Lending Protocol	October 2021	$130 million
PolyNetwork	DeFi Bridge	August 2021	$600 million
KuCoin	Centralized Exchange	September 2020	$281 million
BitGrail	Centralized Exchange	February 2018	$170 million
Coincheck	Centralized Exchange	January 2018	$530 million
Mt. Gox	Centralized Exchange	February 2014	$480 million

Sources: Bloomberg, WSJ, Elliptic Inc. Amounts calculated based on fund values at the time of theft.

Examples of Crypto Collapses

Name	Type of Business	Date of Collapse	Loss Amount
Three Arrows Capital	Hedge Fund	July 2022	$3 billion
Voyager	Lending Firm	July 2022	$1 billion - $10 billion
Celsius	Lending Firm	July 2022	$4.7 billion
Luna + Terra	Blockchain + Stablecoin	March 2022	$45 billion
Coincheck	Centralized Exchange	January 2018	$530 million
Mt. Gox	Centralized Exchange	February 2014	$480 million

Sources: Bloomberg, WSJ, Coinmarketcap.

Examples of Crypto Collapses

Name	Type of Business	Date of Collapse	Loss Amount
Three Arrows Capital	Hedge Fund	July 2022	$3 billion
Voyager	Lending Firm	July 2022	$1 billion - $10 billion
Celsius	**Lending Firm**	**July 2022**	**$4.7 billion**
Luna + Terra	Blockchain + Stablecoin	March 2022	$45 billion
Coincheck	Centralized Exchange	January 2018	$530 million
Mt. Gox	Centralized Exchange	February 2014	$480 million

Sources: Bloomberg, WSJ, Coinmarketcap.

Celsius Collapse

Summary Balance Sheet | Pictured below is the Balance Sheet for Celsius as of August 13, 2021

Celsius - Operating Balance Sheet

As of August 13, 2021

	Assets	Amount ($M)
1	DeFi	$4,483.7
2	Staking	689.6
3	Bank Balances	40.7
4	Undeployed Assets	3,276.2
5	Posted Collateral	2,232.3
6	Institutional Loans	2,241.6
7	CEL Treasury	1,752.6
8	Exchange Balances	2,390.0
9	Mining / Financial Instruments / Other	1,383.6
10	Retail Loans	542.8
11	Undeployable (Prime Trust)	35.9
	Total Assets	**$19,069.0**

	Liabilities & Shareholders' Equity	Amount ($M)
A	Depositor Balances	$12,890.4
B	Depositor Collateral	2,119.7
C	Credit Facility	1,110.4
D	Institutional Collateral	975.8
E	DeFi	785.9
F	Locked CEL	173.6
	Total Liabilities	**$18,055.9**
G	Net Asset Value	$1,013.1
	Total Liabilities and Equity	**$19,069.0**

Source: WSJ, Celsius Investment Memo (September 2021)

Overview of the Talk

- Overview: What is Nakamoto Blockchain
- Nakamoto Blockchain: A Critique in 3 Equations
  - Free Entry Condition (Miners)
  - Incentive Compatibility Condition (Majority Attack)
  - Economic Limits: Flow vs. Stock Problem. Zero Net Attack Cost Theorem.
- Analysis of Double Spending Attacks
- A Way Out: Specialized Capital + Risk of Collapse
  - A Softer Constraint: Stock vs. Stock
  - Collapse Scenarios
- **Open Questions**

Theory Open Question, I

- Open question: are there other ways to generate anonymous, decentralized trust that are less economically limited?
  - 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
  - Suggests $N^*c = p$ is intrinsic to anonymous, decentralized trust
- Vulnerability to majority attack also seems intrinsic to the problem, given CS theory of distributed consensus
- Let's define an attack as cheap if cost is $O(N^*c)$
- Let's define an attack as expensive if cost is $O(N^*C)$
- Question: *is there a blockchain protocol that makes all attacks expensive without reliance on a collapse argument?*

Theory Open Question, I

- Let's first observe that traditional forms of trust solve the problem easily
- Example: collateral + rule-of-law
  - Post C of financial collateral. Lose the collateral if you cheat. Enforced by rule-of-law.
  - Opportunity cost of collateral is rC if the collateral is not used productively
  - Opportunity cost of collateral can be even lower if it can be used productively while locked up (e.g., invested in risk-free bonds)
- So, if rule-of-law works as intended
  - Secure to cheating worth up to C.
  - At cost $\leq rC$.
- Secure a stock at the cost of a flow.

Theory Open Question, I

- Proof of stake and attacks
  - Usual motivation: reduce mining expense and environmental harm
  - Environmental issue is orthogonal to the concerns raised in this paper. Just conceptualize c as per-block opportunity cost of stake
  - But: stakes have *memory*. This may open up new possibilities for making attacks expensive.
- Ethereum Proof-of-Stake
  - In event of a double-spending attack ("finality reversion"): confiscate the attacker's stake ("slashing").
    - Takes advantage of observability of attacker signing conflicting transactions.
    - Takes advantage of memory – stakes are locked up for long enough for the confiscation to work.
  - Makes the cost of double-spending attack a stock not a flow: $\frac{1}{2}N^*C$
- Contrast
  - Bitcoin collapse model: "untargeted slashing". All ASICs have to lose their value for the attack to cost $O(N^*C)$.
  - Ethereum PoS model: "targeted slashing". Only confiscate the attacker's stake. Hence don't need implicit assumption of collapse for security.

Theory Open Question, I

- This is great ... but Ethereum PoS raises new issues not faced by Bitcoin:
  - "Liveness" attacks as opposed to "Safety" attacks
  - These only cost a "flow" not a "stock" in Ethereum's new model
  - Attacker could halt the Ethereum blockchain for long periods of time at low cost
- Leaves me skeptical that Ethereum in current form could be relied upon for global finance.
- And, excited about the open question

Theory 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.

Finance Open Question, I

- 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.

Finance Open Question, II

- More generally, crypto seems a fascinating laboratory through which to study bubbles
- Key observation here: it's a bubble either way!
  - Whether it persists or collapses
- Shiller: naturally occurring Ponzi process
- If a researcher could somehow get data that sheds light on how this phenomenon took off, that would be fascinating

Policy / Legal Theory Open Question

- Anonymous trust strikes me as a real conundrum for policy makers and legal theorists
- There are lots of implicit "legal puts" to the anonymous trust if you look around
  - Ex: if an individual's crypto wallet is stolen by a mugger -> they can call the cops
  - Ex: if a financial institution gets double spent -> they can call the FBI
- So, honest users get some implicit legal protection
- Which enhances the value of the system
- Which provides more cover to black-market users

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: Summary

- Emphasize: model consistent with earliest uses of Bitcoin and blockchain: hobbyists and black market
  - Black market = willing to pay high implicit fees
- 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