Trust at Scale: The Economic Limits of Cryptocurrencies and Blockchains

Eric Budish
University of Chicago, Booth School of Business

Forthcoming,
Quarterly Journal of Economics

Nakamoto's Invention

- Economists have long widely agreed that the market system requires some form of government and rule of law for support
- Uncontroversial among even the most free-market oriented thinkers
  - Smith (1776): "Commerce and manufactures can seldom flourish long in any state" without a legal system, property rights and contract enforcement.
  - Hayek (1960): to maximize freedom, defined as absence of coercion, it is necessary to have a government with the power to coerce
  - Friedman (1962): government sets "rules of the game" and serves as "umpire"

Nakamoto's Invention

- Satoshi Nakamoto (2008) invented a new kind of economic system that does not need the support of government or laws

Nakamoto's Invention

- Satoshi Nakamoto (2008) invented a new kind of economic system that does not need the support of government or laws
- Trust and security instead arise from a combination of cryptography and economic incentives. Completely anonymous and decentralized.
  - CS terminology: "permissionless consensus."
  - Large, anonymous, freely-entering and -exiting set of participants collectively maintains a common data set, without any trusted parties
  - ("a new territory of freedom," "outside the reach of any government")

Nakamoto's Invention

- Satoshi Nakamoto (2008) invented a new kind of economic system that does not need the support of government or laws
- Trust and security instead arise from a combination of cryptography and economic incentives. Completely anonymous and decentralized.
  - CS terminology: "permissionless consensus."
  - Large, anonymous, freely-entering and -exiting set of participants collectively maintains a common data set, without any trusted parties
  - ("a new territory of freedom," "outside the reach of any government")
- Nakamoto's invention enabled cryptocurrencies, including his own, Bitcoin
- The specific data structure maintained is called a blockchain

Nakamoto's Invention

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

Nakamoto's Invention

- Nakamoto's invention captured the world's attention
- Recent peak: $3 trillion
- Even this figure seems to understate the amount of cultural, political and commercial attention that has been paid to blockchains and cryptocurrencies
- Yet, economic usefulness remains an open question
- To date, majority of volume appears speculative, with other widely-documented use case being black market activity (Makarov and Schoar, 2021; Yellen, 2021; Gensler, 2021; Buterin, 2022)

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 activity (Makarov and Schoar, 2021; Yellen, 2021; Gensler, 2021; Buterin, 2022)
- Moreover, most of the speculative volume has been through cryptocurrency exchanges — which are, at least in principle, centralized, trusted financial intermediaries (i.e., not Nakamoto trust in its pure form)

Adam Smith vs. Satoshi Nakamoto

- This paper studies the economics of Nakamoto's novel form of trust.
  - Is it economically viable as an alternative to the traditional market system supported by rule of law?
  - What are the fundamental economics of the fundamental computer science innovation in Nakamoto (2008)?

Adam Smith vs. Satoshi Nakamoto

- This paper studies the economics of Nakamoto's novel form of trust.
  - Is it economically viable as an alternative to the traditional market system supported by rule of law?
  - What are the fundamental economics of the fundamental computer science innovation in Nakamoto (2008)?
- I find that — at least in its pure form, without any implicit protection from rule of law — Nakamoto trust faces serious economic limits.
  - It is unusually expensive in absolute terms relative to the stakes involved
  - Its expense scales linearly with the stakes involved

Adam Smith vs. Satoshi Nakamoto

- This paper studies the economics of Nakamoto's novel form of trust.
  - Is it economically viable as an alternative to the traditional market system supported by rule of law?
  - What are the fundamental economics of the fundamental computer science innovation in Nakamoto (2008)?
- I find that — at least in its pure form, without any implicit protection from rule of law — Nakamoto trust faces serious economic limits.
  - It is unusually expensive in absolute terms relative to the stakes involved
  - Its expense scales linearly with the stakes involved
- Results have an if-then implication: if Nakamoto trust were to become more economically useful, then the costs of securing its trust would become absurd.
  - More than global GDP in some scenarios.

Adam Smith vs. Satoshi Nakamoto

- This paper studies the economics of Nakamoto's novel form of trust.
  - Is it economically viable as an alternative to the traditional market system supported by rule of law?
  - What are the fundamental economics of the fundamental computer science innovation in Nakamoto (2008)?
- I find that — at least in its pure form, without any implicit protection from rule of law — Nakamoto trust faces serious economic limits.
  - It is unusually expensive in absolute terms relative to the stakes involved
  - Its expense scales linearly with the stakes involved
- Results have an if-then implication: if Nakamoto trust were to become more economically useful, then the costs of securing its trust would become absurd.
  - More than global GDP in some scenarios.
- The analysis may also sharpen our conceptual understanding of what is special about traditional forms of trust that are grounded in rule of law and complementary sources (e.g., reputations, relationships, collateral)
  - Key distinction will be economies of scale in the production of trust

The Paper's Argument

- Core of the argument is three equations.

The Paper's Argument

- Core of the argument is three equations.
- Equation (1): zero-profits condition, honest play.
  - The amount of "trust support" devoted to maintaining permissionless consensus reflects the compensation paid by the protocol (work, stake, etc.).

The Paper's Argument

- Core of the argument is three equations.
- Equation (1): zero-profits condition, honest play.
  - The amount of "trust support" devoted to maintaining permissionless consensus reflects the compensation paid by the protocol (work, stake, etc.).
- Equation (2): incentive compatibility condition, majority attack.
  - How much security does a given level of trust support produce?
  - Vulnerability: "majority attack". General issue with distributed consensus—but CS had thought about security as a 0-1 issue, not as an IC constraint.
  - IC: costs of attack must exceed the benefits.

The Paper's Argument

- Core of the argument is three equations.
- Equation (1): zero-profits condition, honest play.
  - The amount of "trust support" devoted to maintaining permissionless consensus reflects the compensation paid by the protocol (work, stake, etc.).
- Equation (2): incentive compatibility condition, majority attack.
  - How much security does a given level of trust support produce?
  - Vulnerability: "majority attack". General issue with distributed consensus—but CS had thought about security as a 0-1 issue, not as an IC constraint.
  - IC: costs of attack must exceed the benefits.
- Together, (1)+(2) imply:
  - (3): recurring, "flow" cost of honest trust support must be large relative to the value of attacking the system
  - Very expensive!
  - Especially as stakes grow! Scales linearly.

The Paper's Argument

- Core of the argument is three equations.
- Equation (1): zero-profits condition, honest play.
  - The amount of "trust support" devoted to maintaining permissionless consensus reflects the compensation paid by the protocol (work, stake, etc.).
- Equation (2): incentive compatibility condition, majority attack.
  - How much security does a given level of trust support produce?
  - Vulnerability: "majority attack". General issue with distributed consensus—but CS had thought about security as a 0-1 issue, not as an IC constraint.
  - IC: costs of attack must exceed the benefits.
- Together, (1)+(2) imply:
  - (3): recurring, "flow" cost of honest trust support must be large relative to the value of attacking the system
  - Very expensive!
  - Especially as stakes grow! Scales linearly.
- Intuition: Nakamoto trust is "memoryless," no scale economies.
- Under idealized attack circumstances, get an even stronger result: "zero net attack cost theorem"

Overview of the Talk

- ~~Overview of Nakamoto (2008) and Related Concepts~~ [See backup slides].
- Nakamoto Trust: A Critique in 3 Equations
- Analysis of Double-Spending Attacks
- A Way Out: Specific Capital + Collapse
- Comparison of Nakamoto Trust and Traditional Trust
- Conclusion

Overview of the Talk

- ~~Overview of Nakamoto (2008) and Related Concepts~~
- **Nakamoto Trust: A Critique in 3 Equations**
- Analysis of Double-Spending Attacks
- A Way Out: Specific Capital + Collapse
- Comparison of Nakamoto Trust and Traditional Trust
- Conclusion

Zero-Profit Condition (Honest Play)

- Conceptual question: how much "trust support" will maintain the permissionless consensus if we restrict all to behave honestly?

Zero-Profit Condition (Honest Play)

- Conceptual question: how much "trust support" will maintain the permissionless consensus if we restrict all to behave honestly?
- N: amount of trust support (computing power, stake)
  - Large finite number of honest participants
  - Follow the permissionless consensus protocol automatically
  - Player i chooses qty of trust support x_i. Define $N = \sum_i x_i$.
  - Eqm concept will be zero-profit. Captures permissionless, free entry/exit.

Zero-Profit Condition (Honest Play)

- Conceptual question: how much "trust support" will maintain the permissionless consensus if we restrict all to behave honestly?
- N: amount of trust support (computing power, stake)
  - Large finite number of honest participants
  - Follow the permissionless consensus protocol automatically
  - Player i chooses qty of trust support x_i. Define $N = \sum_i x_i$.
  - Eqm concept will be zero-profit. Captures permissionless, free entry/exit.
- c: cost per unit time to provide one unit of trust support
  - For proof-of-work, this would include rental cost of capital and variable costs of electricity ($c = rC + \eta$)
  - For proof-of-stake, this would be a rental cost of capital

Zero-Profit Condition (Honest Play)

- Conceptual question: how much "trust support" will maintain the permissionless consensus if we restrict all to behave honestly?
- N: amount of trust support (computing power, stake)
  - Large finite number of honest participants
  - Follow the permissionless consensus protocol automatically
  - Player i chooses qty of trust support x_i. Define $N = \sum_i x_i$.
  - Eqm concept will be zero-profit. Captures permissionless, free entry/exit.
- c: cost per unit time to provide one unit of trust support
  - For proof-of-work, this would include rental cost of capital and variable costs of electricity ($c = rC + \eta$)
  - For proof-of-stake, this would be a rental cost of capital
- p_{block}: compensation per block paid to the player that adds the next block
  - Assume exogenous. Will derive constraints below.
  - Proportional rule: player i wins a given block with prob. $\frac{x_i}{N}$

Zero-Profit Condition (Honest Play)

- D: block difficulty level. Defined as how many units of trust-support-time are needed in expectation to add one new block
- Definition. A *zero-profit honest Nakamoto trust equilibrium* consists of quantities $\{x_i^*\}_{i=1}^I$ and a difficulty level D^* such that participants (i) add one block per unit time (as a normalization), and (ii) earn zero economic profits in expectation.

Zero-Profit Condition (Honest Play)

- D: block difficulty level. Defined as how many units of trust-support-time are needed in expectation to add one new block
- Definition. A *zero-profit honest Nakamoto trust equilibrium* consists of quantities $\{x_i^*\}_{i=1}^I$ and a difficulty level D^* such that participants (i) add one block per unit time (as a normalization), and (ii) earn zero economic profits in expectation.
- Proposition: Let $N^* = \sum_{i=1}^I x_i^*$. In any zero-profit honest Nakamoto trust equilibrium, $D^* = N^*$ and

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

Zero-Profit Condition (Honest Play)

- D: block difficulty level. Defined as how many units of trust-support-time are needed in expectation to add one new block
- Definition. A *zero-profit honest Nakamoto trust equilibrium* consists of quantities $\{x_i^*\}_{i=1}^I$ and a difficulty level D^* such that participants (i) add one block per unit time (as a normalization), and (ii) earn zero economic profits in expectation.
- Proposition: Let $N^* = \sum_{i=1}^I x_i^*$. In any zero-profit honest Nakamoto trust equilibrium, $D^* = N^*$ and

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

- Note: (1) widely known (many prior papers). Standard rent-seeking tournament
- Note: if use Nash eqm, still restrict to honest play, very similar

Incentive Compatibility Condition (Majority Attack)

- Conceptual question: how much security is generated by the amount of honest trust support in (1)?

Incentive Compatibility Condition (Majority Attack)

- Conceptual question: how much security is generated by the amount of honest trust support in (1)?
- Vulnerability: "majority attack"
  - Well-known Achilles' heel of permissionless consensus.
  - Intuition: need majority or super-majority to adjudicate the state if there is a dispute

Incentive Compatibility Condition (Majority Attack)

- Conceptual question: how much security is generated by the amount of honest trust support in (1)?
- Vulnerability: "majority attack"
  - Well-known Achilles' heel of permissionless consensus.
  - Intuition: need majority or super-majority to adjudicate the state if there is a dispute
- In Nakamoto (2008) abstract:
  - "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"

Incentive Compatibility Condition (Majority Attack)

- Conceptual question: how much security is generated by the amount of honest trust support in (1)?
- Vulnerability: "majority attack"
  - Well-known Achilles' heel of permissionless consensus.
  - Intuition: need majority or super-majority to adjudicate the state if there is a dispute
- In Nakamoto (2008) abstract:
  - "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"
- Not surprising. Same issues in 1980's CS literature on what is now called *permissioned* consensus (Lamport et al 1982, Dwork et al 1988)

Incentive Compatibility Condition (Majority Attack)

- Conceptual question: how much security is generated by the amount of honest trust support in (1)?
- Vulnerability: "majority attack"
  - Well-known Achilles' heel of permissionless consensus.
  - Intuition: need majority or super-majority to adjudicate the state if there is a dispute
- In Nakamoto (2008) abstract:
  - "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"
- Not surprising. Same issues in 1980's CS literature on what is now called *permissioned* consensus (Lamport et al 1982, Dwork et al 1988)
- But: with *permissionless* consensus, one has to ask what is an attacker's *incentive* to become a majority
  - In effect, I argue that Nakamoto (2008) made an error conceptualizing security in the same way as the 1980's consensus literature, and not as an IC constraint

Incentive Compatibility Condition (Majority Attack)

- Attack costs
  - Consider an additional player, the attacker, not restricted to honest play.
  - Can attack by choosing AN^* units of trust support, $A > 1$, for an $\frac{A}{A+1}$ majority
  - Cost per unit time: AN^*c
  - Expected duration of attack: $t(A)$. Timing details of attacks will vary by protocol.
  - Call $AN^*c \cdot t(A)$ the gross cost of attack.
  - Attacker can minimize $A \cdot t(A)$: call this $A^* \cdot t(A^*)$

Incentive Compatibility Condition (Majority Attack)

- Attack costs
  - Consider an additional player, the attacker, not restricted to honest play.
  - Can attack by choosing AN^* units of trust support, $A > 1$, for an $\frac{A}{A+1}$ majority
  - Cost per unit time: AN^*c
  - Expected duration of attack: $t(A)$. Timing details of attacks will vary by protocol.
  - Call $AN^*c \cdot t(A)$ the gross cost of attack.
  - Attacker can minimize $A \cdot t(A)$: call this $A^* \cdot t(A^*)$
- Attack benefits
  - 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 the blockchain's economic usefulness grows.

Incentive Compatibility Condition (Majority Attack)

- Attack costs
  - Consider an additional player, the attacker, not restricted to honest play.
  - Can attack by choosing AN^* units of trust support, $A > 1$, for an $\frac{A}{A+1}$ majority
  - Cost per unit time: AN^*c
  - Expected duration of attack: $t(A)$. Timing details of attacks will vary by protocol.
  - Call $AN^*c \cdot t(A)$ the gross cost of attack.
  - Attacker can minimize $A \cdot t(A)$: call this $A^* \cdot t(A^*)$
- Attack benefits
  - 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 the blockchain's economic usefulness grows.
- Definition. Nakamoto trust is *incentive compatible against an outsider attack, on a gross-cost basis*, if the gross cost of attack exceeds the benefits of attack:

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

Incentive Compatibility Condition (Majority Attack)

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

- Remarks

Incentive Compatibility Condition (Majority 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 trust support. Cheaper: as little as $\frac{N^* c}{2}$ per unit time
  - Outside attacker seems more attractive as a conceptual approach. Treats the honest players 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 / stake (Makarov and Schoar; Cong, He and Li)

Incentive Compatibility Condition (Majority 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 trust support. Cheaper: as little as $\frac{N^* c}{2}$ per unit time
  - Outside attacker seems more attractive as a conceptual approach. Treats the honest players 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 / stake (Makarov and Schoar; Cong, He and Li)
- Gross vs. Net Cost
  - (2) is a gross cost. Attacker would earn block rewards for the blocks in their new chain, so Net $<$ Gross. Will come back to this.

Equilibrium Constraint

The Problem

Equilibrium Constraint

The Problem

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

Equilibrium Constraint

The Problem

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

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

Equilibrium Constraint

The Problem

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

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

- Theorem. *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 trust-support providers for maintaining the blockchain ("flow cost of trust") has to be large relative to the benefits of attacking the blockchain

Equilibrium Constraint

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

- Remarks:

Equilibrium Constraint

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

- Remarks:
- Economics: *very expensive* form of trust.
  - 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.
  - Imagine a brand only as trustworthy as its flow investment in advertising.
  - Imagine a country only as secure as its flow expenditure on soldiers at the border.
- Computer 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, laws. Convex returns (analogy: lock on a door)
  - 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
  - Reason 2: Attacker may face frictions relative to honest trust support
  - Reason 3: Attack may harm post-attack value of the cryptocurrency

Net Cost of Attack and a "Zero" Theorem

- Theorem: *if the attack concludes before difficulty adjusts* ($D^{'} = N^{*}$)*, the attacker's cost is the same as honest participants* ($\kappa = 0$)*, and the attack does not cause the value of the cryptocurrency to fall* ($\Delta_{attack} = 0$)*, then the net cost of attack is zero.*

Net Cost of Attack and a "Zero" Theorem

- Theorem: *if the attack concludes before difficulty adjusts ($D' = N^*$), the attacker's cost is the same as honest participants ($\kappa = 0$), and the attack does not cause the value of the cryptocurrency 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 trust-support costs for same reason as honest players are fully compensated for their costs under honest play.

- Gans and Halaburda (2024): generalize the result, give conditions under which the net cost of attack is in fact negative

Graphic Novel Version of the Argument

Traditional Trust

Traditional Trust

Traditional Trust

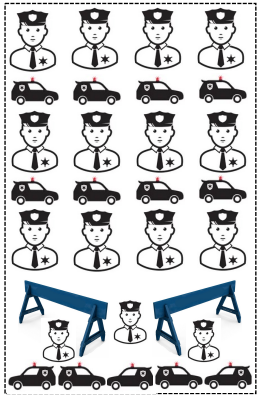
Traditional Trust Model:

Traditional Trust

Traditional Trust Model:

- Security Guards

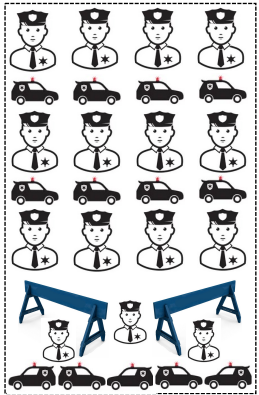
Traditional Trust

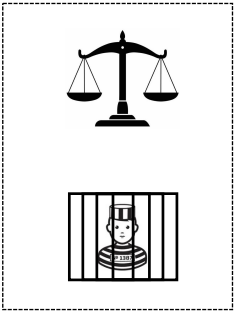


Traditional Trust Model:

- Security Guards
- Police Reinforcements

Traditional Trust





Traditional Trust Model:

- Security Guards
- Police Reinforcements
- Punishment via Rule of Law

Nakamoto Trust (Permissionless Consensus)

Nakamoto Trust (Permissionless Consensus)

Nakamoto Trust (Permissionless Consensus)

Nakamoto Trust (Permissionless Consensus)

Nakamoto Trust Model:

Nakamoto Trust (Permissionless Consensus)

Nakamoto Trust Model:

- Large amount of Security Guards

Nakamoto Trust (Permissionless Consensus)

Nakamoto Trust Model:

- Large amount of Security Guards
- But no additional layers
  (Police, Rule of Law)

Nakamoto Trust (Permissionless Consensus)

Nakamoto Trust Model:

- Large amount of Security Guards
- But no additional layers (Police, Rule of Law)
- So, guards alone must deter attack

Overview of the Talk

- ~~Overview of Nakamoto (2008) and Related Concepts~~
- Nakamoto Trust: A Critique in 3 Equations
- **Analysis of Double-Spending Attacks**
- A Way Out: Specific Capital + Collapse
- Comparison of Nakamoto Trust and Traditional Trust
- Conclusion

Double Spending: Analysis Framework

- Canonical attack is called "double spending." Equation (3) tells us that this possibility places an economic limit on Nakamoto trust:

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

Double Spending: Analysis Framework

- Canonical attack is called "double spending." Equation (3) tells us that this possibility places an economic limit on Nakamoto trust:

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

- Duration of attack: $A^* \cdot t(A^*)$
  - Can compute explicitly for Nakamoto (2008) protocol.
  - Function of escrow period e and # of blocks the attacker double spends for k.

Double Spending: Analysis Framework

- Canonical attack is called "double spending." Equation (3) tells us that this possibility places an economic limit on Nakamoto trust:

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

- Duration of attack: $A^* \cdot t(A^*)$
  - Can compute explicitly for Nakamoto (2008) protocol.
  - Function of escrow period e and # of blocks the attacker double spends for k.
- Benefits of attack: V_{attack}
  - A majority attacker will not double-spend for a cappuccino at Starbucks
  - They will use their majority to conduct transactions that are as large as possible given current uses of cryptocurrencies (potentially, many such transactions)
  - V_{attack}, therefore, can be understood as a statistic on the economic usefulness of Nakamoto trust. ("Max economic throughput")
  - Thought experiment 1: what is the equilibrium cost of security for a given dollar value of V_{attack}?
  - Thought experiment 2: assume $V_{attack} = kV_{honest}$. What is the equilibrium cost of security as a % of honest transaction volume? (Average, not max)

Securing Against an Attack of Size V_{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	5.00%	720%	262,801%	0.0025%
To Secure:				
$1 thousand	$50.0 dollars	$7.2 thousand	$2.6 million	2.5 cents
$1 million	$50.0 thousand	$7.2 million	$2.6 billion	$25.0 dollars
$1 billion	$50.0 million	$7.2 billion	$2.6 trillion	$25.0 thousand
$10 billion	$500.0 million	$72.0 billion	$26.3 trillion	$250.0 thousand
$100 billion	$5.0 billion	$720.0 billion	$262.8 trillion	$2.5 million

Securing Against an Attack of Size V_{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	5.00%	720%	262,801%	0.0025%
To Secure:				
$1 thousand	$50.0 dollars	$7.2 thousand	$2.6 million	2.5 cents
$1 million	$50.0 thousand	$7.2 million	$2.6 billion	$25.0 dollars
$1 billion	$50.0 million	$7.2 billion	$2.6 trillion	$25.0 thousand
$10 billion	$500.0 million	$72.0 billion	$26.3 trillion	$250.0 thousand
$100 billion	$5.0 billion	$720.0 billion	$262.8 trillion	$2.5 million

- Per-block costs follow directly from (3), rewritten as $\frac{p_{block}}{V_{attack}} \geq \frac{1}{At}$

Securing Against an Attack of Size V_{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	5.00%	720%	262,801%	0.0025%
To Secure:				
$1 thousand	$50.0 dollars	$7.2 thousand	$2.6 million	2.5 cents
$1 million	$50.0 thousand	$7.2 million	$2.6 billion	$25.0 dollars
$1 billion	$50.0 million	$7.2 billion	$2.6 trillion	$25.0 thousand
$10 billion	$500.0 million	$72.0 billion	$26.3 trillion	$250.0 thousand
$100 billion	$5.0 billion	$720.0 billion	$262.8 trillion	$2.5 million

- Per-block costs follow directly from (3), rewritten as $\frac{p_{block}}{V_{attack}} \geq \frac{1}{At}$
- Major difficulty: how costs scale with size of attack and over time. $40bn attack requires all of global GDP

Securing Against an Attack of Size V_{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	5.00%	720%	262,801%	0.0025%
To Secure:				
$1 thousand	$50.0 dollars	$7.2 thousand	$2.6 million	2.5 cents
$1 million	$50.0 thousand	$7.2 million	$2.6 billion	$25.0 dollars
$1 billion	$50.0 million	$7.2 billion	$2.6 trillion	$25.0 thousand
$10 billion	$500.0 million	$72.0 billion	$26.3 trillion	$250.0 thousand
$100 billion	$5.0 billion	$720.0 billion	$262.8 trillion	$2.5 million

- Per-block costs follow directly from (3), rewritten as $\frac{p_{block}}{V_{attack}} \geq \frac{1}{At}$
- Major difficulty: how costs scale with size of attack and over time. $40bn attack requires all of global GDP
- % tax sounds more reasonable per transaction, but even tiny tx's have to pay security costs dictated by large attacks

Securing Against an Attack of Size V_{attack}: Sensitivity Analysis

Table 3. Annual Cost to Secure Against Attack: Sensitivity Analysis

	# Blocks of Double Spending + Escrow Period ($k + e$)					
	1	6	12	36	144	1,008
Per-Block Security Costs as % of V_{attack}	26.74%	8.55%	5.00%	1.99%	0.57%	0.09%
Annual Security Costs if V_{attack} Equals:						
\$1 thousand	\$14.1 M	\$4.5 M	\$2.6 M	\$1.0 M	\$301.3 K	\$47.8 K
\$1 million	\$14.1 B	\$4.5 B	\$2.6 B	\$1.0 B	\$301.3 M	\$47.8 M
\$1 billion	\$14.1 T	\$4.5 T	\$2.6 T	\$1.0 T	\$301.3 B	\$47.8 B
\$10 billion	\$140.5 T	\$44.9 T	\$26.3 T	\$10.5 T	\$3.0 T	\$478.2 B
\$100 billion	\$1.4 Q	\$449.1 T	\$262.8 T	\$104.7 T	\$30.1 T	\$4.8 T

Securing Against an Attack of Size V_{attack}: Sensitivity Analysis

Table 3. Annual Cost to Secure Against Attack: Sensitivity Analysis

	# Blocks of Double Spending + Escrow Period ($k + e$)					
	1	6	12	36	144	1,008
Per-Block Security Costs as % of V_{attack}	26.74%	8.55%	5.00%	1.99%	0.57%	0.09%
Annual Security Costs if V_{attack} Equals:						
\$1 thousand	\$14.1 M	\$4.5 M	\$2.6 M	\$1.0 M	\$301.3 K	\$47.8 K
\$1 million	\$14.1 B	\$4.5 B	\$2.6 B	\$1.0 B	\$301.3 M	\$47.8 M
\$1 billion	\$14.1 T	\$4.5 T	\$2.6 T	\$1.0 T	\$301.3 B	\$47.8 B
\$10 billion	\$140.5 T	\$44.9 T	\$26.3 T	\$10.5 T	\$3.0 T	\$478.2 B
\$100 billion	\$1.4 Q	\$449.1 T	\$262.8 T	\$104.7 T	\$30.1 T	\$4.8 T

- Long escrow periods (e.g. $k + e = 1$ day or 1 week) improve the picture by 1-2 orders of magnitude, but costs still very high

Securing Against an Attack of Size V_{attack}: Sensitivity Analysis

Table 3. Annual Cost to Secure Against Attack: Sensitivity Analysis

	# Blocks of Double Spending + Escrow Period ($k + e$)					
	1	6	12	36	144	1,008
Per-Block Security Costs as % of V_{attack}	26.74%	8.55%	5.00%	1.99%	0.57%	0.09%
Annual Security Costs if V_{attack} Equals:						
\$1 thousand	\$14.1 M	\$4.5 M	\$2.6 M	\$1.0 M	\$301.3 K	\$47.8 K
\$1 million	\$14.1 B	\$4.5 B	\$2.6 B	\$1.0 B	\$301.3 M	\$47.8 M
\$1 billion	\$14.1 T	\$4.5 T	\$2.6 T	\$1.0 T	\$301.3 B	\$47.8 B
\$10 billion	\$140.5 T	\$44.9 T	\$26.3 T	\$10.5 T	\$3.0 T	\$478.2 B
\$100 billion	\$1.4 Q	\$449.1 T	\$262.8 T	\$104.7 T	\$30.1 T	\$4.8 T

- Long escrow periods (e.g. $k + e = 1$ day or 1 week) improve the picture by 1-2 orders of magnitude, but costs still very high
- Even the 1-week case requires an annual expense of \$48bn to keep Bitcoin secure up to \$1bn attack.
  - 5% of Global GDP to secure against \$100bn attack.

Costs to Secure Against Attack as % of Honest Volume

- Alternative thought experiment: honest volume is V_{honest} per block, attacker can double-spend this amount for exactly k blocks, i.e., $V_{attack} = kV_{honest}$

Table 4. Costs to Secure Against Attack as % of Honest Volume

V_{attack} # Blocks of Honest Volume (k)	Escrow Period (e)					
	1	6	12	36	144	1,008
1	18.07%	7.61%	4.69%	1.94%	0.57%	0.09%
6	45.68%	30.00%	21.55%	10.45%	3.31%	0.54%
12	56.26%	43.10%	33.85%	18.58%	6.39%	1.08%
36	70.01%	62.67%	55.73%	38.82%	16.78%	3.17%
144	82.02%	79.50%	76.66%	67.14%	43.26%	11.52%
1,008	91.63%	91.20%	90.68%	88.67%	80.63%	46.87%

Costs to Secure Against Attack as % of Honest Volume

- Alternative thought experiment: honest volume is V_{honest} per block, attacker can double-spend this amount for exactly k blocks, i.e., $V_{attack} = kV_{honest}$

Table 4. Costs to Secure Against Attack as % of Honest Volume

V_{attack} # Blocks of Honest Volume (k)	Escrow Period (e)					
	1	6	12	36	144	1,008
1	18.07%	7.61%	4.69%	1.94%	0.57%	0.09%
6	45.68%	30.00%	21.55%	10.45%	3.31%	0.54%
12	56.26%	43.10%	33.85%	18.58%	6.39%	1.08%
36	70.01%	62.67%	55.73%	38.82%	16.78%	3.17%
144	82.02%	79.50%	76.66%	67.14%	43.26%	11.52%
1,008	91.63%	91.20%	90.68%	88.67%	80.63%	46.87%

Costs to Secure Against Attack as % of Honest Volume

- Alternative thought experiment: honest volume is V_{honest} per block, attacker can double-spend this amount for exactly k blocks, i.e., $V_{attack} = kV_{honest}$

Table 4. Costs to Secure Against Attack as % of Honest Volume

V_{attack} # Blocks of Honest Volume (k)	Escrow Period (e)					
	1	6	12	36	144	1,008
1	18.07%	7.61%	4.69%	1.94%	0.57%	0.09%
6	45.68%	30.00%	21.55%	10.45%	3.31%	0.54%
12	56.26%	43.10%	33.85%	18.58%	6.39%	1.08%
36	70.01%	62.67%	55.73%	38.82%	16.78%	3.17%
144	82.02%	79.50%	76.66%	67.14%	43.26%	11.52%
1,008	91.63%	91.20%	90.68%	88.67%	80.63%	46.87%

Double Spending Attack: Takeaways

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

- Consistent with modest early use cases of Bitcoin (computer parts, silk road, online gambling)—if double-spending worth \$1k, then cost per tx just \$0.025
- Consistent with larger-scale black-market uses of Bitcoin—users willing to pay high tx costs. (Ex: \$1000 per tx secures up to \$40M in base case)
- Casts doubt on Bitcoin / Nakamoto trust as major component of mainstream global financial system (too expensive!)

Double Spending Attack: Takeaways

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

- Consistent with modest early use cases of Bitcoin (computer parts, silk road, online gambling)—if double-spending worth \$1k, then cost per tx just \$0.025
- Consistent with larger-scale black-market uses of Bitcoin—users willing to pay high tx costs. (Ex: \$1000 per tx secures up to \$40M in base case)
- Casts doubt on Bitcoin / Nakamoto trust as major component of mainstream global financial system (too expensive!)
- Surprises to the CS community:
  1. for the system to be secure for large transactions requires tx costs that are ridiculous for small transactions
  2. that a long-enough escrow period isn't enough
- Source of both surprises: missed eqm reasoning that one needs to worry about larger and larger attacks if Bitcoin / Nakamoto trust gets more economically useful. Security is not 0-1, but more like a (very high) % tax.

Overview of the Talk

- ~~Overview of Nakamoto (2008) and Related Concepts~~
- Nakamoto Trust: A Critique in 3 Equations
- Analysis of Double-Spending Attacks
- **A Way Out: Specific Capital + Collapse**
- Comparison of Nakamoto Trust and Traditional Trust
- Conclusion

Specific Capital + Collapse

- 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
- Proof-of-stake: capital is specialized by construction.

Specific Capital + Collapse

- 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
- Proof-of-stake: capital is specialized by construction.
- If (i) capital is specialized, and (ii) attack causes collapse of its value, 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: $O(N^* C)$
- This will be my candidate answer to the "Chicago lunch table question"

Analysis if Attacker Loses their Capital Value

- Let $c = rC + \eta$, where C = fixed cost of capital, r = interest rate per unit time, η = variable costs.
- Honest equilibrium (1) can be written as:

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

- Outside attacker needs N^*C of capital. Assume attacker loses all of their capital value. IC constraint 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:

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

Analysis if Attacker Loses their Capital Value

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

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

- MUCH more secure than before, because of r which is an interest rate per unit time.

Analysis if Attacker Loses their Capital Value

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

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

- MUCH more secure than before, because of r which is an interest rate per unit time. Here is an example calculation for Bitcoin:
- Assume the capital share of mining is $\mu = 0.4$ (De Vries, 2018, 2022)
- Assume annual discount rate for ASICs is 50% (depreciation and risk), which implies that the per-unit-time disount rate is $r \approx 0.0008\%$. So $\frac{r}{\mu} = 0.002\%$.

Analysis if Attacker Loses their Capital Value

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

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

- MUCH more secure than before, because of r which is an interest rate per unit time. Here is an example calculation for Bitcoin:
- Assume the capital share of mining is $\mu = 0.4$ (De Vries, 2018, 2022)
- Assume annual discount rate for ASICs is 50% (depreciation and risk), which implies that the per-unit-time disount rate is $r \approx 0.0008\%$. So $\frac{r}{\mu} = 0.002\%$.
- Compare to $\frac{1}{A^* \cdot t(A^*)}$ factor on the right-hand-side of (3) from before. Base case value was $\frac{1}{A^* \cdot t(A^*)} = 5.00\%$
- So 2500-fold improvement in the cost of security.

Analysis if Attacker Loses their Capital Value

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

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

- MUCH more secure than before, because of r which is an interest rate per unit time. Here is an example calculation for Bitcoin:
- Assume the capital share of mining is $\mu = 0.4$ (De Vries, 2018, 2022)
- Assume annual discount rate for ASICs is 50% (depreciation and risk), which implies that the per-unit-time disount rate is $r \approx 0.0008\%$. So $\frac{r}{\mu} = 0.002\%$.
- Compare to $\frac{1}{A^* \cdot t(A^*)}$ factor on the right-hand-side of (3) from before. Base case value was $\frac{1}{A^* \cdot t(A^*)} = 5.00\%$
- So 2500-fold improvement in the cost of security.
- (N.B. these same values for μ and r and $p_{block} = \$250k$ imply capital stock of \$12.5B, which about matches retail prices.)

Issue: How Exactly does Attacker Lose their Capital?

How might an attacker lose their capital value N^*C? Several possibilities, each with problems:

Issue: How Exactly does Attacker Lose their Capital?

How might an attacker lose their capital value N^*C? Several possibilities, each with problems:

1. Collapse of the cryptocurrency.
   - Works, but (i) harms honest participants, (ii) raises possiblility of a sabotage attack.
   - Value of sabotage likely grows with economic usefulness / financial market integration.

Issue: How Exactly does Attacker Lose their Capital?

How might an attacker lose their capital value N^*C? Several possibilities, each with problems:

1. Collapse of the cryptocurrency.
   - Works, but (i) harms honest participants, (ii) raises possiblility of a sabotage attack.
   - Value of sabotage likely grows with economic usefulness / financial market integration.
2. Internal-to-protocol punishment without collapse
   - This is what Ethereum proof-of-stake with "slashing" is trying to accomplish
   - But, bumps up against impossibility theorems (Tas et al, 2023; Budish, Lewis-Pye and Roughgarden, 2024)
   - Intuition: a large-enough attacker can prevent the protocol from punishing them, controls the legal system ($\rho = \frac{2}{3}$ the cutoff)

Issue: How Exactly does Attacker Lose their Capital?

How might an attacker lose their capital value N^*C? Several possibilities, each with problems:

1. Collapse of the cryptocurrency.
   - Works, but (i) harms honest participants, (ii) raises possiblility of a sabotage attack.
   - Value of sabotage likely grows with economic usefulness / financial market integration.
2. Internal-to-protocol punishment without collapse
   - This is what Ethereum proof-of-stake with "slashing" is trying to accomplish
   - But, bumps up against impossibility theorems (Tas et al, 2023; Budish, Lewis-Pye and Roughgarden, 2024)
   - Intuition: a large-enough attacker can prevent the protocol from punishing them, controls the legal system ($\rho = \frac{2}{3}$ the cutoff)
3. External-to-protocol punishment from rule of law
   - Certainly works ... but the whole question of this paper is whether Nakamoto trust is economically viable without government and rule of law.

Overview of the Talk

- ~~Overview of Nakamoto (2008) and Related Concepts~~
- Nakamoto Trust: A Critique in 3 Equations
- Analysis of Double-Spending Attacks
- A Way Out: Specific Capital + Collapse
- **Comparison of Nakamoto Trust and Traditional Trust**
- Conclusion

Beckerian Deterrence as an Economy of Scale

- For concreteness, consider a financial transaction between two parties of size V, where one of the parties can cheat and steal the other's assets.
- Party 1: choose from $\{Engage, Don't\ Engage\}$
- Party 2: choose from $\{Honest,\ Cheat\}$
- Payoffs

Beckerian Deterrence as an Economy of Scale

- For concreteness, consider a financial transaction between two parties of size V, where one of the parties can cheat and steal the other's assets.
- Party 1: choose from $\{Engage, Don't\ Engage\}$
- Party 2: choose from $\{Honest,\ Cheat\}$
- Payoffs
  - Party 1 plays *Engage*, Party 2 plays *Honest*: both parties get a payoff of $b > 0$

Beckerian Deterrence as an Economy of Scale

- For concreteness, consider a financial transaction between two parties of size V, where one of the parties can cheat and steal the other's assets.
- Party 1: choose from $\{Engage, Don't\ Engage\}$
- Party 2: choose from $\{Honest,\ Cheat\}$
- Payoffs
  - Party 1 plays *Engage*, Party 2 plays *Honest*: both parties get a payoff of $b > 0$
  - Party 1 plays *Engage*, Party 2 plays *Cheat*: Party 2 gets $+V$ and Party 1 gets $-V$

Beckerian Deterrence as an Economy of Scale

- For concreteness, consider a financial transaction between two parties of size V, where one of the parties can cheat and steal the other's assets.
- Party 1: choose from $\{Engage, Don't\ Engage\}$
- Party 2: choose from $\{Honest,\ Cheat\}$
- Payoffs
  - Party 1 plays *Engage*, Party 2 plays *Honest*: both parties get a payoff of $b > 0$
  - Party 1 plays *Engage*, Party 2 plays *Cheat*: Party 2 gets $+V$ and Party 1 gets $-V$
  - Party 1 plays *Don't Engage*: both parties get 0.

Beckerian Deterrence as an Economy of Scale

- For concreteness, consider a financial transaction between two parties of size V, where one of the parties can cheat and steal the other's assets.
- Party 1: choose from $\{Engage, Don't\ Engage\}$
- Party 2: choose from $\{Honest,\ Cheat\}$
- Payoffs
  - Party 1 plays *Engage*, Party 2 plays *Honest*: both parties get a payoff of $b > 0$
  - Party 1 plays *Engage*, Party 2 plays *Cheat*: Party 2 gets $+V$ and Party 1 gets $-V$
  - Party 1 plays *Don't Engage*: both parties get 0.
- Eqm. One-shot game, as described: clearly only equilibrium is *Don't Engage*.

Beckerian Deterrence as an Economy of Scale

- Now add a legal system with the power to enforce contracts:

Beckerian Deterrence as an Economy of Scale

- Now add a legal system with the power to enforce contracts:
- If Party 2 cheats, Party 1 can pay a cost c_l to bring the matter to court.
- The court can perfectly observe Party 2's play, and can compel Party 2 to return Party 1's assets and pay a fine of $f > 0$
  - Payoffs in this scenario:
  - Party 1's payoff is $-c_l$ and Party 2's payoff is $-f$.

Beckerian Deterrence as an Economy of Scale

- Now add a legal system with the power to enforce contracts:
- If Party 2 cheats, Party 1 can pay a cost c_l to bring the matter to court.
- The court can perfectly observe Party 2's play, and can compel Party 2 to return Party 1's assets and pay a fine of $f > 0$
  - Payoffs in this scenario:
  - Party 1's payoff is $-c_l$ and Party 2's payoff is $-f$.
- Eqm. Clearly, legal system makes it an equilibrium to transact honestly. The credible threat of enforcement deters Player 2 from cheating.

Beckerian Deterrence as an Economy of Scale

- Now add a legal system with the power to enforce contracts:
- If Party 2 cheats, Party 1 can pay a cost c_l to bring the matter to court.
- The court can perfectly observe Party 2's play, and can compel Party 2 to return Party 1's assets and pay a fine of $f > 0$
  - Payoffs in this scenario:
  - Party 1's payoff is $-c_l$ and Party 2's payoff is $-f$.
- Eqm. Clearly, legal system makes it an equilibrium to transact honestly. The credible threat of enforcement deters Player 2 from cheating.
- Observe: on path, the court need not even involve itself with the transaction.
- This is the economy of scale for traditional trust: *A society that pays a fixed cost of operating a court system can facilitate honest transactions that have zero marginal cost of security because of the deterrence effect.*

Complementary Sources of Trust: Reputations, Relationships, Collateral

- Similar scale economies of trust arise in the private sector from fixed-cost investments in brands, reputations, relationships (Nelson, 1974; Fudenberg, Levine and Maskin, 1994; Tadelis, 1999; Baker, Gibbons and Murphy, 2002; Levin, 2003).

- Remark 1: These often work in conjunction with rule of law.
  - E.g., can't directly copy a brand's look-and-feel; lower-bound payoffs in Levin 2003.

- Remark 2: Collateral a particularly important example for finance.
  - Cost of general-purpose collateral as a source of trust support is *zero* under the assumptions of the Modigliani-Miller theorem.
  - And note collateral works in conjunction with rule-of-law.

Comparison of Trust Models

	Cost to Break Trust	Variable Cost of Honest Play	Fixed Cost Investments that Support Honest Play
Nakamoto Trust in its original form (Natamoto, 2008)	Flow. Attacker pays cost of honest trust support $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	–
Repeated Interaction (Schelling, 1956; Aumann, 1959)	Stock. Attacker loses the net present value of the trust relationship $V_{trust} = \frac{1}{1-\delta}\tau$.	0, given pre-existing trust relationship.	Investment in trust relationship, brand, reputation.
Credible Deterrence (Hayek, 1960; Becker, 1968)	Arbitrarily high up to value of life and assets. Attacker can be punished by the state.	"Security guards" in Figure 1. Variable costs of monitoring and enforcement associated with specific locations or transactions.	Investment in government and legal system.
Collateral	Stock + Punishment. Attacker loses the value of their collateral plus any additional punishment.	0, under conditions of Modigliani-Miller theorem. Requires legal enforcement mechanism which entails fixed costs.	Collateral value.
Nakamoto with Collapse	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	ASICs or other specialized computational equipment.
Proof-of-Stake in its original form	Flow. Attacker pays opportunity cost of capital $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	Cryptocurrency stake.
Proof-of-Stake with Slashing (Idealized)	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	Cryptocurrency stake.

Comparison of Trust Models

	Cost to Break Trust	Variable Cost of Honest Play	Fixed Cost Investments that Support Honest Play
Nakamoto Trust in its original form (Natamoto, 2008)	Flow. Attacker pays cost of honest trust support $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	–
Repeated Interaction (Schelling, 1956; Aumann, 1959)	Stock. Attacker loses the net present value of the trust relationship $V_{trust} = \frac{1}{1-\delta}\tau$.	0, given pre-existing trust relationship.	Investment in trust relationship, brand, reputation.
Credible Deterrence (Hayek, 1960; Becker, 1968)	Arbitrarily high up to value of life and assets. Attacker can be punished by the state.	"Security guards" in Figure 1. Variable costs of monitoring and enforcement associated with specific locations or transactions.	Investment in government and legal system.
Collateral	Stock + Punishment. Attacker loses the value of their collateral plus any additional punishment.	0, under conditions of Modigliani-Miller theorem. Requires legal enforcement mechanism which entails fixed costs.	Collateral value.
Nakamoto with Collapse	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	ASICs or other specialized computational equipment.
Proof-of-Stake in its original form	Flow. Attacker pays opportunity cost of capital $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	Cryptocurrency stake.
Proof-of-Stake with Slashing (Idealized)	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	Cryptocurrency stake.

Comparison of Trust Models

	Cost to Break Trust	Variable Cost of Honest Play	Fixed Cost Investments that Support Honest Play
Nakamoto Trust in its original form (Natamoto, 2008)	Flow. Attacker pays cost of honest trust support $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	–
Repeated Interaction (Schelling, 1956; Aumann, 1959)	Stock. Attacker loses the net present value of the trust relationship $V_{trust} = \frac{1}{1-\delta}\tau$.	0, given pre-existing trust relationship.	Investment in trust relationship, brand, reputation.
Credible Deterrence (Hayek, 1960; Becker, 1968)	Arbitrarily high up to value of life and assets. Attacker can be punished by the state.	"Security guards" in Figure 1. Variable costs of monitoring and enforcement associated with specific locations or transactions.	Investment in government and legal system.
Collateral	Stock + Punishment. Attacker loses the value of their collateral plus any additional punishment.	0, under conditions of Modigliani-Miller theorem. Requires legal enforcement mechanism which entails fixed costs.	Collateral value.
Nakamoto with Collapse	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	ASICs or other specialized computational equipment.
Proof-of-Stake in its original form	Flow. Attacker pays opportunity cost of capital $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	Cryptocurrency stake.
Proof-of-Stake with Slashing (Idealized)	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	Cryptocurrency stake.

Comparison of Trust Models

	Cost to Break Trust	Variable Cost of Honest Play	Fixed Cost Investments that Support Honest Play
Nakamoto Trust in its original form (Natamoto, 2008)	Flow. Attacker pays cost of honest trust support $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	–
Repeated Interaction (Schelling, 1956; Aumann, 1959)	Stock. Attacker loses the net present value of the trust relationship $V_{trust} = \frac{1}{1-\delta}\tau$.	0, given pre-existing trust relationship.	Investment in trust relationship, brand, reputation.
Credible Deterrence (Hayek, 1960; Becker, 1968)	Arbitrarily high up to value of life and assets. Attacker can be punished by the state.	"Security guards" in Figure 1. Variable costs of monitoring and enforcement associated with specific locations or transactions.	Investment in government and legal system.
Collateral	Stock + Punishment. Attacker loses the value of their collateral plus any additional punishment.	0, under conditions of Modigliani-Miller theorem. Requires legal enforcement mechanism which entails fixed costs.	Collateral value.
Nakamoto with Collapse	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	ASICs or other specialized computational equipment.
Proof-of-Stake in its original form	Flow. Attacker pays opportunity cost of capital $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	Cryptocurrency stake.
Proof-of-Stake with Slashing (Idealized)	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	Cryptocurrency stake.

Comparison of Trust Models

	Cost to Break Trust	Variable Cost of Honest Play	Fixed Cost Investments that Support Honest Play
Nakamoto Trust in its original form (Natamoto, 2008)	Flow. Attacker pays cost of honest trust support $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	–
Repeated Interaction (Schelling, 1956; Aumann, 1959)	Stock. Attacker loses the net present value of the trust relationship $V_{trust} = \frac{1}{1-\delta}\tau$.	0, given pre-existing trust relationship.	Investment in trust relationship, brand, reputation.
Credible Deterrence (Hayek, 1960; Becker, 1968)	Arbitrarily high up to value of life and assets. Attacker can be punished by the state.	"Security guards" in Figure 1. Variable costs of monitoring and enforcement associated with specific locations or transactions.	Investment in government and legal system.
Collateral	Stock + Punishment. Attacker loses the value of their collateral plus any additional punishment.	0, under conditions of Modigliani-Miller theorem. Requires legal enforcement mechanism which entails fixed costs.	Collateral value.
Nakamoto with Collapse	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	ASICs or other specialized computational equipment.
Proof-of-Stake in its original form	Flow. Attacker pays opportunity cost of capital $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	Cryptocurrency stake.
Proof-of-Stake with Slashing (Idealized)	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	Cryptocurrency stake.

Comparison of Trust Models

	Cost to Break Trust	Variable Cost of Honest Play	Fixed Cost Investments that Support Honest Play
Nakamoto Trust in its original form (Natamoto, 2008)	Flow. Attacker pays cost of honest trust support $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	–
Repeated Interaction (Schelling, 1956; Aumann, 1959)	Stock. Attacker loses the net present value of the trust relationship $V_{trust} = \frac{1}{1-\delta}\tau$.	0, given pre-existing trust relationship.	Investment in trust relationship, brand, reputation.
Credible Deterrence (Hayek, 1960; Becker, 1968)	Arbitrarily high up to value of life and assets. Attacker can be punished by the state.	"Security guards" in Figure 1. Variable costs of monitoring and enforcement associated with specific locations or transactions.	Investment in government and legal system.
Collateral	Stock + Punishment. Attacker loses the value of their collateral plus any additional punishment.	0, under conditions of Modigliani-Miller theorem. Requires legal enforcement mechanism which entails fixed costs.	Collateral value.
Nakamoto with Collapse	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	ASICs or other specialized computational equipment.
Proof-of-Stake in its original form	Flow. Attacker pays opportunity cost of capital $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	Cryptocurrency stake.
Proof-of-Stake with Slashing (Idealized)	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	Cryptocurrency stake.

Comparison of Trust Models

	Cost to Break Trust	Variable Cost of Honest Play	Fixed Cost Investments that Support Honest Play
Nakamoto Trust in its original form (Natamoto, 2008)	Flow. Attacker pays cost of honest trust support $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	–
Repeated Interaction (Schelling, 1956; Aumann, 1959)	Stock. Attacker loses the net present value of the trust relationship $V_{trust} = \frac{1}{1-\delta}\tau$.	0, given pre-existing trust relationship.	Investment in trust relationship, brand, reputation.
Credible Deterrence (Hayek, 1960; Becker, 1968)	Arbitrarily high up to value of life and assets. Attacker can be punished by the state.	"Security guards" in Figure 1. Variable costs of monitoring and enforcement associated with specific locations or transactions.	Investment in government and legal system.
Collateral	Stock + Punishment. Attacker loses the value of their collateral plus any additional punishment.	0, under conditions of Modigliani-Miller theorem. Requires legal enforcement mechanism which entails fixed costs.	Collateral value.
Nakamoto with Collapse	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	ASICs or other specialized computational equipment.
Proof-of-Stake in its original form	Flow. Attacker pays opportunity cost of capital $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	Cryptocurrency stake.
Proof-of-Stake with Slashing (Idealized)	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	Cryptocurrency stake.

Comparison of Trust Models

	Cost to Break Trust	Variable Cost of Honest Play	Fixed Cost Investments that Support Honest Play
Nakamoto Trust in its original form (Natamoto, 2008)	Flow. Attacker pays cost of honest trust support $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	–
Repeated Interaction (Schelling, 1956; Aumann, 1959)	Stock. Attacker loses the net present value of the trust relationship $V_{trust} = \frac{1}{1-\delta}\tau$.	0, given pre-existing trust relationship.	Investment in trust relationship, brand, reputation.
Credible Deterrence (Hayek, 1960; Becker, 1968)	Arbitrarily high up to value of life and assets. Attacker can be punished by the state.	"Security guards" in Figure 1. Variable costs of monitoring and enforcement associated with specific locations or transactions.	Investment in government and legal system.
Collateral	Stock + Punishment. Attacker loses the value of their collateral plus any additional punishment.	0, under conditions of Modigliani-Miller theorem. Requires legal enforcement mechanism which entails fixed costs.	Collateral value.
Nakamoto with Collapse	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	ASICs or other specialized computational equipment.
Proof-of-Stake in its original form	Flow. Attacker pays opportunity cost of capital $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	Cryptocurrency stake.
Proof-of-Stake with Slashing (Idealized)	Stock. Attacker loses specialized capital worth $N^* C$.	p_{block} per period.	Cryptocurrency stake.

Sense of Magnitudes: Traditional Trust vs. Nakamoto Trust

- Sense of magnitudes for finance

Sense of Magnitudes: Traditional Trust vs. Nakamoto Trust

- Sense of magnitudes for finance
  - Total annual spending in the US on police, prisons and courts is about $300bn.
  - Real value added in the U.S. financial industry is about $800bn. (see Philippon, 2015)
  - So $1 trillion is a conservative upper bound for the cost of trust in the U.S. financial sector (former includes much non-finance, latter includes much non-trust).

Sense of Magnitudes: Traditional Trust vs. Nakamoto Trust

- Sense of magnitudes for finance
  - Total annual spending in the US on police, prisons and courts is about $300bn.
  - Real value added in the U.S. financial industry is about $800bn. (see Philippon, 2015)
  - So $1 trillion is a conservative upper bound for the cost of trust in the U.S. financial sector (former includes much non-finance, latter includes much non-trust).
  - Transaction volume in U.S. finance easily exceeds $1 quadrillion per year (Budish and Sunderam, 2024)

Sense of Magnitudes: Traditional Trust vs. Nakamoto Trust

- Sense of magnitudes for finance
  - Total annual spending in the US on police, prisons and courts is about $300bn.
  - Real value added in the U.S. financial industry is about $800bn. (see Philippon, 2015)
  - So $1 trillion is a conservative upper bound for the cost of trust in the U.S. financial sector (former includes much non-finance, latter includes much non-trust).
  - Transaction volume in U.S. finance easily exceeds $1 quadrillion per year (Budish and Sunderam, 2024)
- So we can conservatively upper bound $\frac{F}{V_{honest}} + c$ for the financial sector by 0.1%

Sense of Magnitudes: Traditional Trust vs. Nakamoto Trust

- Sense of magnitudes for finance
  - Total annual spending in the US on police, prisons and courts is about $300bn.
  - Real value added in the U.S. financial industry is about $800bn. (see Philippon, 2015)
  - So $1 trillion is a conservative upper bound for the cost of trust in the U.S. financial sector (former includes much non-finance, latter includes much non-trust).
  - Transaction volume in U.S. finance easily exceeds $1 quadrillion per year (Budish and Sunderam, 2024)
- So we can conservatively upper bound $\frac{F}{V_{honest}} + c$ for the financial sector by 0.1%
- Many fees in traditional finance, especially for large transactions, are 0.01% or less of transaction volume.
- Not meant to be an apology for traditional finance (see Greenwood and Scharfstein, 2013; Philippon, 2015; Zingales, 2015). But comparison with Nakamoto trust is night-and-day.

Overview of the Talk

- ~~Overview of Nakamoto (2008) and Related Concepts~~
- Nakamoto Trust: A Critique in 3 Equations
- Analysis of Double-Spending Attacks
- A Way Out: Specific Capital + Collapse
- Comparison of Nakamoto Trust and Traditional Trust
- **Conclusion**

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*
- Permissionless consensus has to date been of limited usefulness in its pure form
- My work suggests this is intrinsic to the technology

Conclusion: Summary

- Anonymous, decentralized trust enabled by Nakamoto (2008) blockchain: *ingenious but expensive*
- Permissionless consensus has to date been of limited usefulness in its pure form
- My work suggests this is intrinsic to the technology
- Emphasize: model consistent with continued use for black-market activity
  - Black market = willing to pay high implicit fees

Conclusion: Summary

- Anonymous, decentralized trust enabled by Nakamoto (2008) blockchain: *ingenious but expensive*
- Permissionless consensus has to date been of limited usefulness in its pure form
- My work suggests this is intrinsic to the technology
- Emphasize: model consistent with continued use for black-market activity
  - Black market = willing to pay high implicit fees
- Also emphasize: analysis is consistent with the usefulness of the blockchain data structure without Nakamoto's novel form of trust ("permissioned blockchain" or "distributed ledger technology")
  - Example: Central Bank Digital Currencies (CBDCs) take some technical inspiration from cryptocurrencies but are anchored in traditional trust from rule of law and the reputation of central banks

Conclusion: Summary

- Anonymous, decentralized trust enabled by Nakamoto (2008) blockchain: *ingenious but expensive*
- Permissionless consensus has to date been of limited usefulness in its pure form
- My work suggests this is intrinsic to the technology
- Emphasize: model consistent with continued use for black-market activity
  - Black market = willing to pay high implicit fees
- Also emphasize: analysis is consistent with the usefulness of the blockchain data structure without Nakamoto's novel form of trust ("permissioned blockchain" or "distributed ledger technology")
  - Example: Central Bank Digital Currencies (CBDCs) take some technical inspiration from cryptocurrencies but are anchored in traditional trust from rule of law and the reputation of central banks
- 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 is the source of its economic limits

Directions for Future Research

- Direction 1: Is there a "solution" to this paper's critique of Bitcoin and Nakamoto trust?

Directions for Future Research

- Direction 1: Is there a "solution" to this paper's critique of Bitcoin and Nakamoto trust?
- Informally: is there a way to generate trust in a public dataset that has some of the anonymity and decentralization aspects of Nakamoto while being significantly less economically constrained by the arguments in this paper

Directions for Future Research

- Direction 1: Is there a "solution" to this paper's critique of Bitcoin and Nakamoto trust?
- Informally: is there a way to generate trust in a public dataset that has some of the anonymity and decentralization aspects of Nakamoto while being significantly less economically constrained by the arguments in this paper
- Many responses to my paper are described in Appendix A (first circulated June 2018)
- The most promising responses combine blockchain-based trust with some external source of trust support
  - Ex: Ethereum proof-of-stake with slashing, works up to $\rho = \frac{2}{3}$. Needs external punishment above $\frac{2}{3}$.
  - Ex: "digital gold" arguments (university endowments, Blackrock Bitcoin ETF)

Directions for Future Research

- Direction 1: Is there a "solution" to this paper's critique of Bitcoin and Nakamoto trust?
- Informally: is there a way to generate trust in a public dataset that has some of the anonymity and decentralization aspects of Nakamoto while being significantly less economically constrained by the arguments in this paper
- Many responses to my paper are described in Appendix A (first circulated June 2018)
- The most promising responses combine blockchain-based trust with some external source of trust support
  - Ex: Ethereum proof-of-stake with slashing, works up to $\rho = \frac{2}{3}$. Needs external punishment above $\frac{2}{3}$.
  - Ex: "digital gold" arguments (university endowments, Blackrock Bitcoin ETF)
- This in turn raises the next question

Directions for Future Research

- Direction 2: Multi-layered trust.
- Traditional trust is often "multi-layered": law, reputations, relationships, brands, collateral, technology, etc., often working together in the same transaction, without even drawing much notice
  - Ex: JPMorgan combination of reputation, collateral, laws
  - Ex: local coffee shop (reputation, relational contracts, food safety laws)

Directions for Future Research

- Direction 2: Multi-layered trust.
- Traditional trust is often "multi-layered": law, reputations, relationships, brands, collateral, technology, etc., often working together in the same transaction, without even drawing much notice
  - Ex: JPMorgan combination of reputation, collateral, laws
  - Ex: local coffee shop (reputation, relational contracts, food safety laws)
- Q1: How do we model the production possibilities frontier for trust?
- Q2: Does permissionless consensus expand the frontier? Are there combinations of permissionless consensus protocols + traditional sources of trust that are superior to existing alternatives?
  - Ex: Ethereum PoS with slashing to punish small attackers and rule-of-law to deter large attackers. How does this compare to law all the way?
  - How does this compare to JPMorgan's trust bundle?

Backup Slides

What is Nakamoto Blockchain (1/4)

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

Sender	Receiver	Amount	Signature
Alice	Bob	$10	*Alice*

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

What is Nakamoto Blockchain (1/4)

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

Sender	Receiver	Amount	Signature
Alice	Bob	$10	*Alice*

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

What is Nakamoto Blockchain (1/4)

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

Sender	Receiver	Amount	Signature
Alice	Bob	$10	*Alice*

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

What is Nakamoto Blockchain (2/4)

Nakamoto (2008) Blockchain Innovation

What is Nakamoto Blockchain (2/4)

Nakamoto (2008) Blockchain Innovation

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

What is Nakamoto Blockchain (2/4)

Nakamoto (2008) Blockchain Innovation

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

What is Nakamoto Blockchain (2/4)

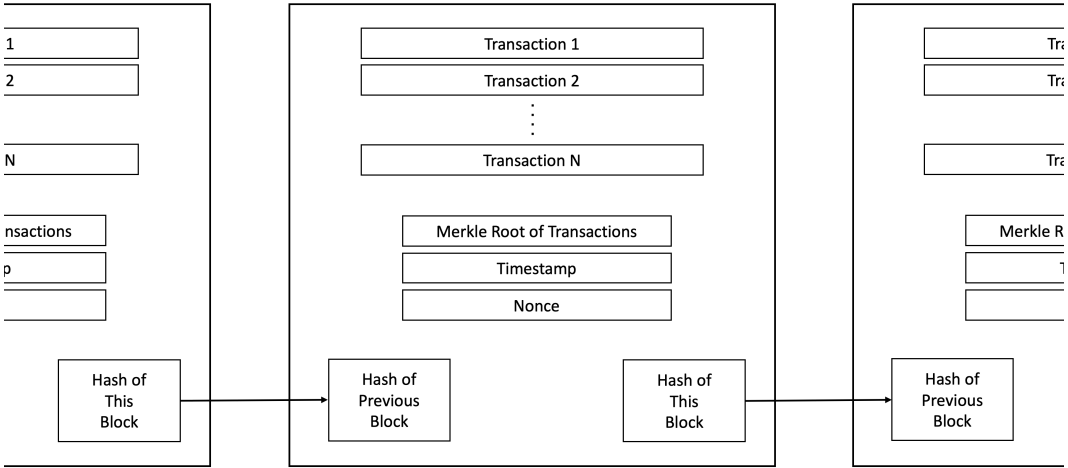
Nakamoto (2008) Blockchain Innovation

- **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)

What is Nakamoto Blockchain (2/4)

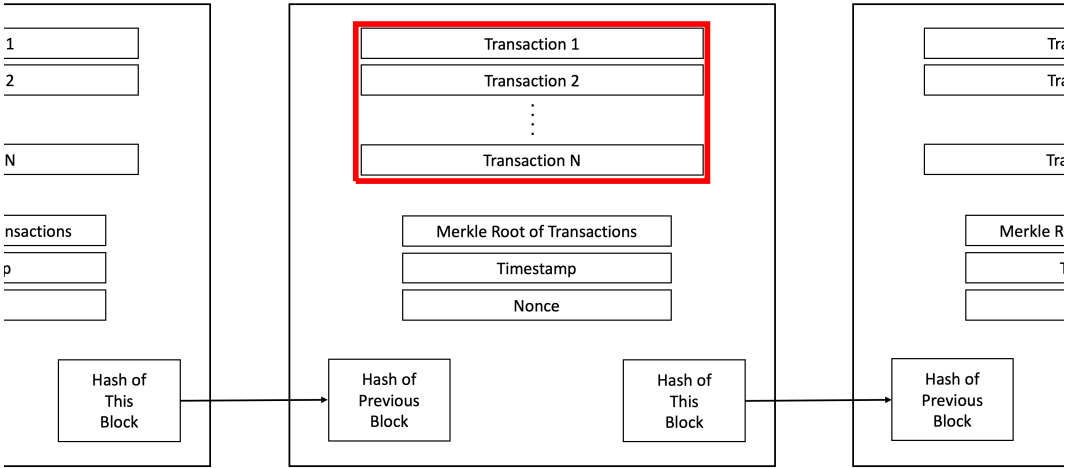
Nakamoto (2008) Blockchain Innovation

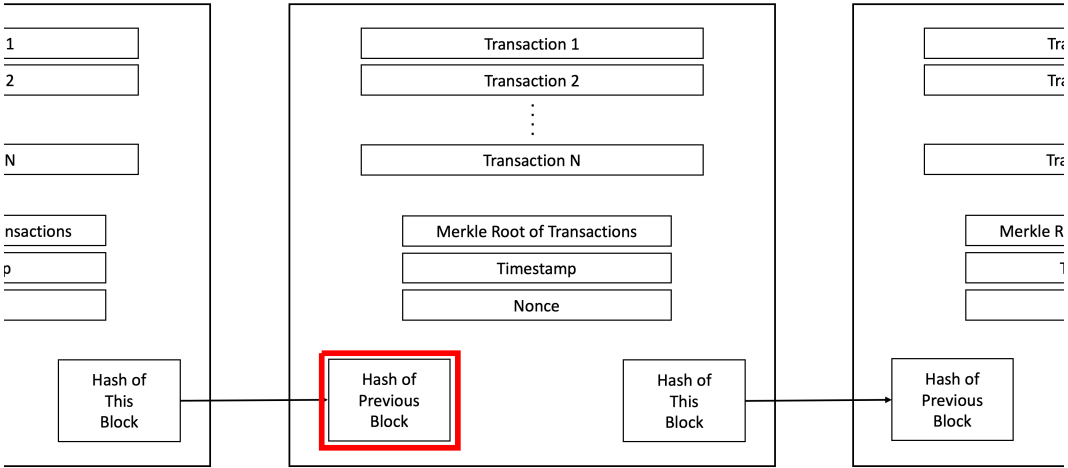
- **I: Pending Transactions List**
  - Users submit transactions to a pending transactions list, called mempool
  - Like a google spreadsheet — not considered official yet
- **II: Valid Blocks**
  - Any computer around the world can compete for the right to add transactions from the mempool to a data 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.

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

$$000000000000000000008b6f6fb83f8d74512ef1e0af29e642dd20daddd7d318f$$

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

$$000000000000000000008b6f6fb83f8d74512ef1e0af29e642dd20daddd7d318f$$

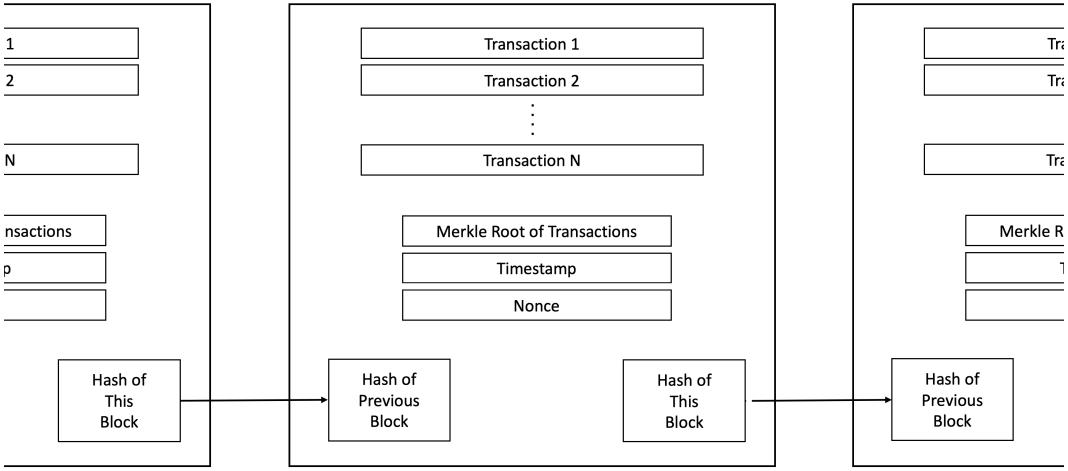
  - Called "proof of work" – hard to find, easy to check.
  - Winner is compensated with newly issued Bitcoins ("mining") and optional transaction fees (see Huberman, Leshno and Moallemi 2021)
  - Tournament difficulty adjusts every two weeks, calibrated to take about 10 minutes

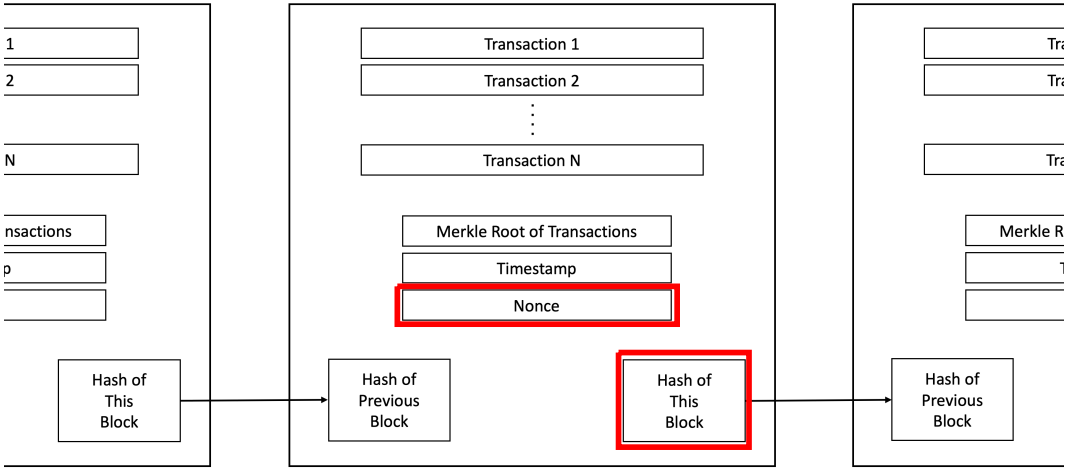
SHA-256 Hash Function: Example

Name	SHA256 Hash	Name	SHA256 Hash
Nikhil Agarwal	534dab9b320debd919af5c902a1863ba7e2e9a28997ab09407be0a4754310f6	Patrick Agte	793f05f67a13ab6f15c8e08bd2d91b478cc80c105f3a180f30b02f87c400d820
Claudia Allende	50f2a93835480197a4d9b6407242e48713bbd856a71dfb9fb9397e204f85da85	Daniel Aronoff	3bc90109fc906b8663b793bb6081f9c15b335709afdbe8c309081cb1a4dacb81
Lawrence Ausubel	f951cda93b6cd470992226c29540d46f35f96d49fdbb3094b93b9ceca85cc131	Ian S. Ball	a5bc5ce3054612five8af983c9779dbfa632683a8da54fe5aa81e9f85b9cf4e0fb
Martino Banchio	a2fe70e1f172657b7f56621ad55f9b02a249a957b99b1f339d420d818e728285	Dirk Bergemann	35e9b858bdfbf6dc8230298dabb6251bfed7c402971e4a9e1666d295e3400928
Eric Budish	222667cd0f3bb9c5baf1be47ed5f416f6cd2b413c0721a45ad6bae5977867c16	Juan Camilo Castillo	0ab4c824573f82286a8d7cf68204e12ec38d02a5c383835d08349aff82244168
Oguzhan Celebi	2bda967c9d50a6f194404d16cc7bf3a32eecc984bd8e5b2993109a7b71c1411c	Alex Chan	58f3a5d47e59e1071155956fa2abdd200ec6acd8be2fb84a8d2d04836172d19
Daniel Chen	4aea1aabadbc812454ca74e6dbfe1ac230b21878be14505ab39d1c34dc5f38a7	Peter Cramton	1bf69ceecc7b693529b9987c30429e1176ab15471a166d990237082816810ca91
Luciano I. de Castro	6190e2c96ce94fd7414bfd35fde4761dddd21bb2bdcf574c2809213371df28af	David Delacretaz	a4ef8e22724a06983d504c5047a2d4008be6d63827cd7e341a865ed5a3c05cc4
Wouter Dessein	798b1ce13d029e0442138592f0941753ffb4abecc77be34ecf051007106365c1	Laura Doval	cdc39475e56c674291a10beec7657e699727a6c1fb63c6d4036d1eb6a0ed0e4e
Jeremy T. Fox	d7080b5a442d7f55eb5d7aa5a1f8cd87e0f2cc7720294aabf1aaed50dfb09732	Drew Fudenberg	cf4d1c521f7b762ac6d80986227e259356d3bd5326ec3bbfe4a3a45412f4eeb	
Diego Gentile Passaro	5c1d006ae7d557019208a0d7e8e639ddbe4c12b19dc9617dbcfaf78a11202ccb	Yannai A. Gonczarowski	b74591c331c6f8a2a5f0054118286100 5c99662e093f54c20acc807f2a3ed63
Alexander Haberman	cda48aea98472ecc327a87516be0bd605aa68cc454b386fed34519239a75ff15	Guillaume Haeringer	cd91749ff949699f2a25f5c1dc6926239a7186a624fc7c7cf45bc24c7c07550
Dong Woo Hahm	061d8bdddaf1fb74022a931e0c2fcb2e0246a7e8707774a6c8d8c6e51d8e3adc	YingHua He	abd8a67d3d67db74a265d8a9a818bedd615eeb0597f50bd4e311d09ab4ff59a1
Clemence M. Idoux	b5a018630f1773b7d9d2a7f50571074a8a60ca753489a7ffad0db0db51780474	Nicole Immorlica	5f0539de379fd53c85c57d4a874e988fd995a17027a31d23e077000ef15d3b43
Ravi Jagadeesan	714fa62c238c88a1558f9dc5cd9a86aa2d0934466030e9ed553cc4ba7ec09a28	Zi Yang Kang	d1654d7a6b22ce3d5d5222eb904c5ba5019525b8467f037 78c5e98f66163a37d
Adam Kapor	0fb39f066629908f13419dd1d0cd251aeaf81666a3cd7114ba75d5f4e5696e814e	Navin Kartik	85f06b56461ee6a8105fb40d590ffe36751c34ec37db52d94f7d5594428bf774
Jakub Kastl	985c369f70241c3274edc5e3a7669355cc3d1031a11f862c2191a9c2319ce0e9	Judd B. Kessler	03da2b35489614c6f8c82fa5c32c0bd3e91e67054c6f524a57de1cfc58b0ec44
Jacob A. Leshno	ccfcf7bf7ec9390e0cad3ad7ab7bdb599edc05ff2a704f79d0154a085c14825b	Kwok Hao Lee	048c26b7e83c7f6c22e6b8d42209ac4aaf53fcfce8985978e40a75f2b6a0967
Jacob D. Leshno	01bd9ab65ee828760a2a609813a4753cbfed80f1d349a98deb3bae57e82c1699	Hao Li	0fadb5d98f675f36af215749086dcc8deb3aa77c37e5bd7a0420ef2fca8b3350
Shengwu Li	62f63590404ce5685054d868095a3e958294d8133f708f085716d87541b9e863c	Irene Y. Lo	eed5b47e6e946a4c3b0e8a5d688130adc0beaca8b9edad3a66c6c1b79022de70
Brendan Lucier	d9718a2be14660259f31b6ea737b26f425c9f58f81843dce576d67469c373fa	Hongyao Ma	02f77df6ca202fa577a4ffbeaa01da203e337c2d0e4beb037363346673af9a21
Stephen Morris	c06e28b6b4aaff7d112ac6f8a3f2c6ad88f093628f563d5ecc63aee3305f1ec4	Ellen Muir	e456bd9f5e68b580b773b9770846d68ab295ef4f9277a6f2ea0b3cdd366bbad5
Christopher Neilson	e83c3888fa6e0969823 65ef8650d4584e297cc0fdb6ad9a600174c0f393bfc70	Fernando Ochoa	6eaa5ae7de95e602b0daeb82eddb4a0e651f1022000a9dd2d0a0c072d41b5b48
Michael Ostrovsky	b7bdbdf42ecbe53b2e296e22ef9d1d1623de256e6b702c317980767a7e518b56	Bobak Pakzad-Hurson	dda782a93600045b67ee039e630e5af75ca5aef21b35716db03987200b203800
Parag A. Pathak	c80213f73539106531edcb265ba65b94fe6daed3787 9db4da312bca8788c6e35	Daniel Quint	f3c08c5b3222a8acf520b9f7210e0fae782eed5121644039d3ecaaa01cd0dd57
Alex Rees-Jones	e40322f2281fa2ba31ad427610bced94890b42e30bcc4895ccfee812b9c83353	Eric Richert	8d39b8c92a80b91a0262c70b4dc3b0f0d8973e85ca0266336604e05987d9ddd6
Marzena Rostek	0e68efa4ab8cbbb0e0b597199dd20e3543e6506086 41a884fc6fca350a739cdbe	Anna Russo	f04ba03a5692f8cd561195b81da26143ec9751a64094c864ae4f12f3ba4f017a3
Tobias Salz	4b3b784be4c54e5dc6cc56c05e93d0bbb12d4ba3c842b1495d0614fb61b31b1	Tayfun Sönmez	df70c49f6a9b873c9ee2b8899a230238286aadd116bd1855bece5a80eff1db54
Alexander Teytelboym	db2da11117c1fc1662090fee690083c15bf2b9d966ba29b241b8e74b0778a72	Juuso Toikka	6c18efab6056df6836ef9bfe7a12f982dc76696cba3b17c7dc44b46833ac36d6
M. Utku Ünver	be21c48db12dbb2551c67706f1e4cb39e18d983897f1134788d9ed6bff50f7	Quitze Valenzuela-Stookey	bf8dd3e74d84c462a67e996ccdad0c6c197c5ef381c1d2c20cc8480a3839d452

SHA-256 Hash Function: Example

Name	SHA256 Hash	Name	SHA256 Hash
Nikhil Agarwal	534dab9b320debd919af5c902a1863ba7e2e9a28997ab09407be0a4754310916	Patrick Agte	793f05f67a13ab6f15c8e08bd2d91b478cc80c105f3a180f30b02f87c400d820
Claudia Allende	50f2a93835480197a4d9b6407242e48713bbd856a71dfb9fb9397e204f85da85	Daniel Aronoff	3bc90109fc906b8663b793bb6081f9c15b335709afdbe8c309081cb1a4dacb81
Lawrence Ausubel	f951cda93b6cd470992226c29540d46f35f96d49fdbb3094b93b9ceca85cc131	Ian S. Ball	a5bc5ce3054612516af983c9779dbfa632683a8da54fe5aa81e9f85b9cf4e0fb
Martino Banchio	a2fe70e1f172657b7f56621ad55f9b02a249a957b99b1f339d420d818e728285	Dirk Bergemann	35e9b858bdfbf6dc8230298dabb6251bfed7c402971e4a9e1666d295e3400928
Eric Budish	222667cd0f3bb9c5baf1be47ed5f416f6cd2b413c0721a45ad6bae5977867c16	Juan Camilo Castillo	0ab4c824573f82286a8d7cf68204e12ec38d02a5c383835d08349aff82244168
Oguzhan Celebi	2bda967c9d50a6f194404d16cc7bf3a32eecc984bd8e5b2993109a7b71c1411c	Alex Chan	58f3a5d47e59e1071155956ffa2abdd200ec6ac8be2fb84a8d2d04836172d19
Daniel Chen	4aea1aabadbc812454ca74e6dbfe1ac230b21878be14505ab39d1c34dc5f38a7	Peter Cramton	1bf69ceecc7b693529b9987c30429e1176ab15471a166d99023708281861dca91
Luciano I. de Castro	6190e2c96ce94fd7414bfd35fde4761dddd21bb2bdcf574c2809213371df28af	David Delacretaz	a4ef8e22724a06983d504c5047a2d4008be6d63827cd7e341a865ed5a3c05cc4
Wouter Dessein	798b1ce13d029e0442138592f0941753ffba4becc77be34ef051007106365c1	Laura Doval	cdc39475e56c674291a10beec7657e699727a6c1fb63cfd46036d1eb6a0ed04e
Jeremy T. Fox	d7080b5a442d7f55eb5d7aa5a1f8cd87e0f2cc7720294aabf1aaed50dfb09732	Drew Fudenberg	cf4d1c521f7b762ac6d80986227e259356d3bd5326ec3bbfe4a3a45412f4eeb	
Diego Gentile Passaro	5c1d006ae7d557019208a0d7e8e639ddbe4c12b19dc9617dbcfaf78a11202ccb	Yannai A. Gonczarowski	b74591c331c6f8a2a5f005411828610005c99662e093f54c20acc807f2a3ed63
Alexander Haberman	cda48aea98472ecc327a87516be0bd605aa68cc454b386fed34519239a75ff15	Guillaume Haeringer	cd91749ff949699f92a25f5c1dc6926239a7186a624fc7c7cf45ac24c7c07550
Dong Woo Hahm	061d8bdddaf1fb74022a931e0c2fcb2e0246a7e8707774a6c8d8c6e51d8e3adc	YingHua He	abd8a67d3d67db74a265d8a9a818bedd615eeb0597f50bd4e311d09ab4ff59a1
Clemence M. Idoux	b5a018630f1773b7d9d2a7f50571074a8a60ca753489a7ffad0db0db51780474	Nicole Immorlica	5f0539de379fd53c85c57d4a874e988fd995a17027a31d23e077000ef15d3b43
Ravi Jagadeesan	714fa62c238c88a1558f9dc5cd9a86aa2d0934466030e9ed553cc4ba7ec09a28	Zi Yang Kang	d1654d7a6b22cd3e35d5222eb904c5ba50195025b846f703778c5e98f66163a37d
Adam Kapor	0fb39f066299087f13419dd1d0cd251aeaf81666a3cd7114ba75d5f4e5696e814e	Navin Kartik	85f06b56461ee6a8105fb40d590ffe367514ec37db52d94f7d5594428bf774
Jakub Kastl	985c369f70241c3274edc5e3a7669355cc3d1031a11f862c2191a9c2319ce0e9	Judd B. Kessler	03da2b35489614c6f8c82fa5c32c0bd3e91e67054c6f524a57de1cfc58b0ec44
John Lazarev	ccfcf7bf7ec9390e0cad3ad7ab7bdb599edc05ff2a704f79d0145ed85c14825b	Kwok Hao Lee	048c26b7e83c7f6c22e6b8d42209ac4aaf53fcfce8985978e40a75f2b6a0967
Jacob D. Leshno	01bd9ab65ee828760a2a6098134a4753cbfed80f1d349a98deb3bae57e82c1699	Hao Li	0fadb5d98f675f36af215749086dcc8deb3aa77c37e5bd7a0420ef2fca8b3350
Shengwu Li	62f63590404ce5685054d86895a3e958294d8133f708f085716d87541b9e863c	Irene Y. Lo	eed5b47e6e946a4c3b0e8a5d688130adc0beaca8b9edad3a66c6c1b79022de70
Brendan Lucier	d9718a2be14660259f31b6ea737b26f425c9f58f81843dce576d67469c373fa	Hongyao Ma	02f77df6ca202fa577a4ffbeaa01d203e337c2d0e4beb037363346673af9a21
Stephen Morris	c06e28b6b4aaff7d112ac6f8a3f2c6ad88f093628f563d5ecc63aee3305f1ec4	Ellen Muir	e456bd9f5e68b580b773b9770846d68ab295ef4f9277a6f2ea0b3cdd366bbad5
Christopher Neilson	e83c3888fa6e096982365ef8650d4584e297cc0fdb6ad9a600174c0f393bfc70	Fernando Ochoa	6eaa5ae7de95e602b0daeb82eddb4a0e651f1022000a9dd2d0a0c072d41b5b48
Michael Ostrovsky	b7bdbdf42ecbe53b2e296a22ef9d1d1623de256e6b702c317980767a7e518b56	Bobak Pakzad-Hurson	dda782a93600045b67ee039e630e5af75ca5aef21b35716db03987200b203800
Parag A. Pathak	c80213f73539106531edcb265ba65b94fe6daed37879db4da312bca8788c6e35	Daniel Quint	f3c08c5b322a8acf520b9f7210e0fae782eed5121644039d3ecaaa01cd0dd57
Alex Rees-Jones	e40322f2281fa2ba31ad427610bced94890b42e30bcc4895ccfee812b9c83353	Eric Richert	8d39b8c92a80b91a0262c70b4dc3b0f0d8973e85ca0266336604e05987d9ddd6
Marzena Rostek	0e68efa4ab8cbb0e0b597199dd20e3543e6506068641a884fc6fca350a739cdbe	Anna Russo	f04ba03a5692f8cd561195b81da26143ec9751a64094c864aef12f3ba4f017a3
Tobias Salz	4b3b784be4c54e5d6cc56c05e93d00bbb12d4ba3c842b1495d0614fb61b31b1	Tayfun Sönmez	df70c49f6a9b873c9ee2b8899a230238286aadd116bd1855bece5a80eff1db54
Alexander Teytelboym	db2da11117c1fc1662090fee690083c15bf2b9d966ba29b241b8e74b0778a72	Juuso Toikka	6c18efab6056df636ef9bfe7a12f982dc76696cba3b17c7dc44b46833ac36d6
M. Utku Ünver	be21c48db12dbb2551c677036f1e4cb39e18d983897f1134788d9ed6bff50f7	Quitze Valenzuela-Stookey	bf8dd3e74d84c462a67e996ccdad0c6c197c5ef381c1d2c20cc8480a3839d452


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 $729,999$**:**
- **Hash:** $0000000000000000008b6f6fb83f8d745\ldots$

What is Nakamoto Blockchain (4/4)

- **IV Longest-Chain Convention**
  - Once a miner finds a lucky alphanumeric string, all participants are supposed to move on to 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*

What is Nakamoto Blockchain (4/4)

- **IV Longest-Chain Convention**
  - Once a miner finds a lucky alphanumeric string, all participants are supposed to move on to 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))

What is Nakamoto Blockchain (4/4)

- **IV Longest-Chain Convention**
  - Once a miner finds a lucky alphanumeric string, all participants are supposed to move on to 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: vulnerable to attack by a 51% majority. Can outpace honest miners with probability one.
    - (Not surprising that it is vulnerable. Distributed consensus that pre-dates Nakamoto also vulnerable – e.g. Lamport et al 1982, Dwork et al 1988)

What is Nakamoto Blockchain: Summary

- From the Nakamoto (2008) abstract:

What is Nakamoto Blockchain: Summary

- From the Nakamoto (2008) abstract:

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

What is Nakamoto Blockchain: Summary

- From the Nakamoto (2008) abstract:

  "We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an 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)

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
- A very interesting open question is whether the blockchain data structure is economically valuable in contexts where the trust is grounded in traditional sources (Budish and Sunderam, 2024). Will return to this at the end.

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

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

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 are not memory-less*:* they are locked up on chain (like collateral) and observably persist over time (like reputation). This opens up the possibility of punishing attackers by confiscating their stakes, making attacks more expensive.
  - Recent research shows that this approach to security, while intuitively compelling, only partially works
  - Issue is, roughly, how do you punish the attacker if the attacker is large enough to adjudicate the truth (Tas et al, 2023; Budish, Lewis-Pye, and Roughgarden 2024)

Directions for Future Research

- ("Don't be against the future. Be for the future.")
- An innovation I'd like to see:
  - Ethereum-style PoS with slashing (to make attacks $< \frac{2}{3}$ expensive)
  - Either permissionless or "gently permissioned"
  - Explicit rule-of-law protection (for $\geq \frac{2}{3}$ attackers)
  - Ability for law enforcement to de-anonymize transactions with subpoena
  - (Can't have your cake and eat it too. If rule-of-law providing security against large attacks, should also be able to enforce its laws against ordinary crime).

Examples of 51% Attacks

Name	Date of First Attack	Amount Stolen	Length of Largest Reorganization
Bitcoin SV	8/3/2021	Unknown	14 Blocks
	6/24/2021	Unknown	Unknown
Verge	2/15/2021	Unknown	560,000 Blocks
	5/22/2018	$1.8 million	NA
	4/4/2018	$1 million	NA
Æternity	12/3/2020	$2.9 million	Unknown
Grin	11/8/2020	Unknown	Unknown
Ethereum Classic	8/29/2020	Unknown	7,000 Blocks
	8/6/2020	$1.7 million	4,200 Blocks
	7/29/2020	$5.6 million	3,700 Blocks
	1/5/2019	$1.1 million	Unknown
Bitcoin Gold	1/23/2020	$100 thousand	29 Blocks
	5/16/2018	$18 million	22 Blocks
Firo	1/18/2019	$5 million	300 Blocks
Vertcoin	12/2/2018	$100 thousand	307 Blocks
Zencash	6/2/2018	$700 thousand	38 Blocks
Litecoin Cash	5/30/2018	Unknown	Unknown
Monacoin	5/13/2018	$90 thousand	Unknown

Sources: Bloomberg, Coindesk, Bitcoinist, CCN, Cointelegraph, bitquery, GitHub Gist and Medium. 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	Date of First Attack	Amount Stolen	Length of Largest Reorganization
Bitcoin SV	**8/3/2021**	**Unknown**	**14 Blocks**
	6/24/2021	**Unknown**	**Unknown**
Verge	2/15/2021	Unknown	560,000 Blocks
	5/22/2018	$1.8 million	NA
	4/4/2018	$1 million	NA
Æternity	12/3/2020	$2.9 million	Unknown
Grin	11/8/2020	Unknown	Unknown
Ethereum Classic	**8/29/2020**	**Unknown**	**7,000 Blocks**
	8/6/2020	**$1.7 million**	**4,200 Blocks**
	7/29/2020	**$5.6 million**	**3,700 Blocks**
	1/5/2019	**$1.1 million**	**Unknown**
Bitcoin Gold	**1/23/2020**	**$100 thousand**	**29 Blocks**
	5/16/2018	**$18 million**	**22 Blocks**
Firo	1/18/2019	$5 million	300 Blocks
Vertcoin	12/2/2018	$100 thousand	307 Blocks
Zencash	6/2/2018	$700 thousand	38 Blocks
Litecoin Cash	5/30/2018	Unknown	Unknown
Monacoin	5/13/2018	$90 thousand	Unknown

Sources: Bloomberg, Coindesk, Bitcoinist, CCN, Cointelegraph, bitquery, GitHub Gist and Medium. 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.

Attacks of Crypto Financial Entities

Name	Type of Business	Date of Attack	Amount Stolen	Attack Vector
Euler Finance	Decentralized Lending Firm	January 2023	$197	Flashloan Attack + Flawed Code
Mango Market	Decentralized Exchange	October 2022	$100 million	Price Manipulation
BNB Chain	DeFi Bridge	October 2022	$568 million	Flawed Code
Wintermute	DeFi Market Maker	September 2022	$160 million	Compromised Wallet Generator
Nomad	DeFi Bridge	August 2022	$200 million	Flawed Code
Horizon Bridge	DeFi Bridge	July 2022	$100	Compromised Private Keys + Governance Control
Beanstalk Farms	DeFi Stablecoin	April 2022	$182 million	Flashloan Attack + Governance Control
Ronin Network	DeFi Bridge	March 2022	$625 million	Compromised Private Keys + Governance Control
Wormhole	DeFi Bridge	February 2022	$320 million	Flawed Code
Qubit Finance	Lending Firm	January 2022	$80	Flawed Code
BitMart	Centralized Exchange	December 2021	$150 million	Compromised Private Keys
C.r.e.a.m. Finance	DeFi Lending Protocol	October 2021	$130 million	Flashloan Attack + Price Manipulation
PolyNetwork	DeFi Bridge	August 2021	$600 million	Flawed Code
KuCoin	Centralized Exchange	September 2020	$281 million	Compromised Private Keys
BitGrail	Centralized Exchange	February 2018	$170 million	Unknown
Coincheck	Centralized Exchange	January 2018	$530 million	Unknown
The DAO	Decentralized Venture Capital	Juny 2016	$55 million	Flawed Code
Mt. Gox	Centralized Exchange	February 2014	$480 million	Compromised Private Keys

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

Attacks of Crypto Financial Entities

Name	Type of Business	Date of Attack	Amount Stolen	Attack Vector
Euler Finance	Decentralized Lending Firm	January 2023	$197	Flashloan Attack + Flawed Code
Mango Market	Decentralized Exchange	October 2022	$100 million	Price Manipulation
BNB Chain	DeFi Bridge	October 2022	$568 million	Flawed Code
Wintermute	DeFi Market Maker	September 2022	$160 million	Compromised Wallet Generator
Nomad	DeFi Bridge	August 2022	$200 million	Flawed Code
Horizon Bridge	DeFi Bridge	July 2022	$100	Compromised Private Keys + Governance Control
Beanstalk Farms	**DeFi Stablecoin**	**April 2022**	**$182 million**	**Flashloan Attack + Governance Control**
Ronin Network	DeFi Bridge	March 2022	$625 million	Compromised Private Keys + Governance Control
Wormhole	DeFi Bridge	February 2022	$320 million	Flawed Code
Qubit Finance	Lending Firm	January 2022	$80	Flawed Code
BitMart	Centralized Exchange	December 2021	$150 million	Compromised Private Keys
C.r.e.a.m. Finance	DeFi Lending Protocol	October 2021	$130 million	Flashloan Attack + Price Manipulation
PolyNetwork	DeFi Bridge	August 2021	$600 million	Flawed Code
KuCoin	Centralized Exchange	September 2020	$281 million	Compromised Private Keys
BitGrail	Centralized Exchange	February 2018	$170 million	Unknown
Coincheck	Centralized Exchange	January 2018	$530 million	Unknown
The DAO	Decentralized Venture Capital	Juny 2016	$55 million	Flawed Code
Mt. Gox	Centralized Exchange	February 2014	$480 million	Compromised Private Keys

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

Collapses of Crypto Financial Entities

Name	Type of Business	Date of Collapse	Entity Size (or Loss Amt)
Genesis	Lending Firm	January 2023	$1 billion - $10 billion
BlockFi	Lending Firm	November 2022	$1 billion - $10 billion
FTX	Centralized Exchange	November 2022	$32 billion
Three Arrows Capital	Hedge Fund	July 2022	$3 billion
Voyager	Lending Firm	July 2022	$1 billion - $10 billion
Celsius	Lending Firm	July 2022	$4 billion - $19 billion
Terra + Luna	Blockchain + Stablecoin	March 2022	$40 billion
Coincheck	Centralized Exchange	January 2018	$530 million (loss amt)
Mt. Gox	Centralized Exchange	February 2014	$480 million (loss amt)

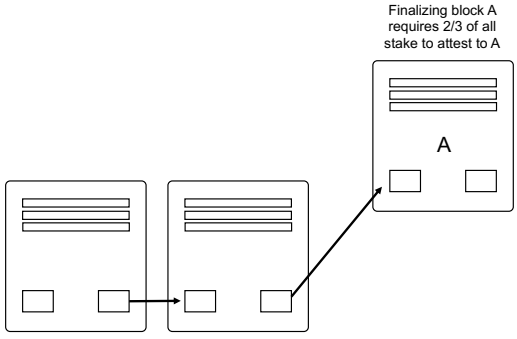
Sources: Bloomberg, WSJ, Coinmarketcap.

Collapses of Crypto Financial Entities

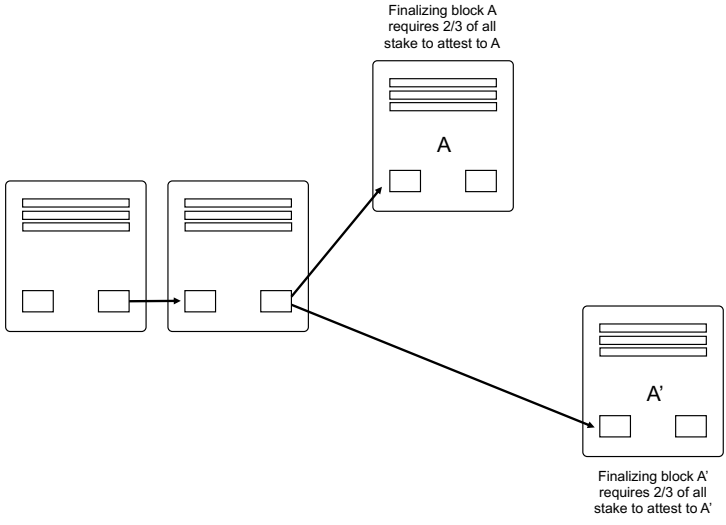
Name	Type of Business	Date of Collapse	Entity Size (or Loss Amt)
Genesis	Lending Firm	January 2023	$1 billion - $10 billion
BlockFi	Lending Firm	November 2022	$1 billion - $10 billion
FTX	**Centralized Exchange**	**November 2022**	**$32 billion**
Three Arrows Capital	Hedge Fund	July 2022	$3 billion
Voyager	Lending Firm	July 2022	$1 billion - $10 billion
Celsius	**Lending Firm**	**July 2022**	**$4 billion - $19 billion**
Terra + Luna	Blockchain + Stablecoin	March 2022	$40 billion
Coincheck	Centralized Exchange	January 2018	$530 million (loss amt)
Mt. Gox	Centralized Exchange	February 2014	$480 million (loss amt)

Sources: Bloomberg, WSJ, Coinmarketcap.

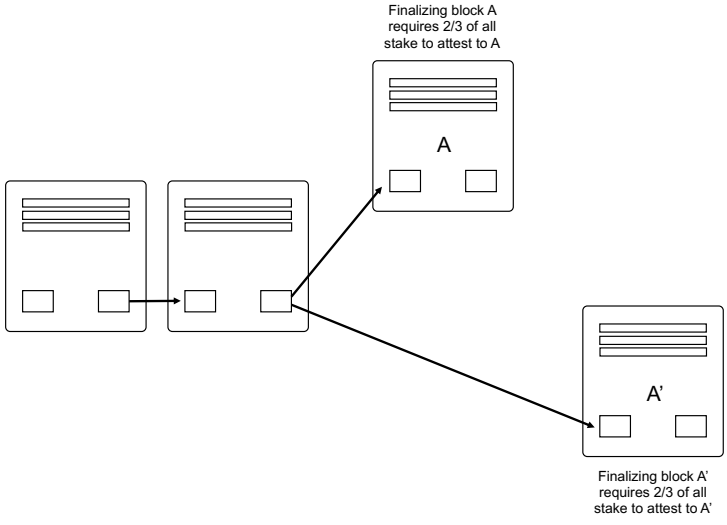
Ethereum PoS: Punishing a Double-Spend Attacker



Ethereum PoS: Punishing a Double-Spend Attacker

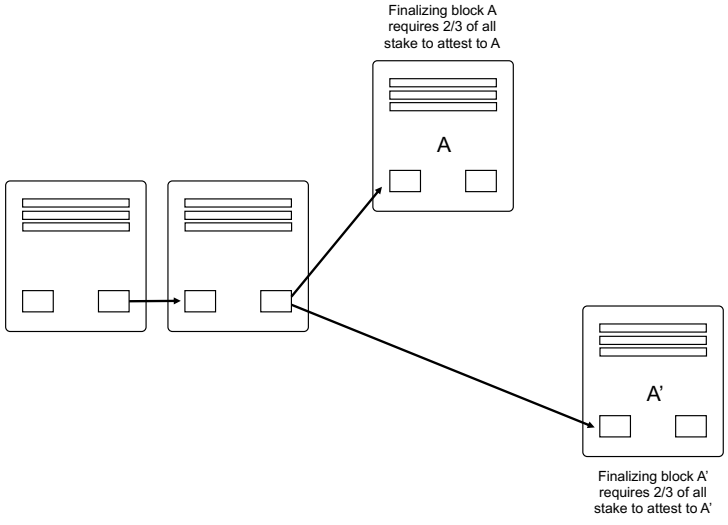


Ethereum PoS: Punishing a Double-Spend Attacker



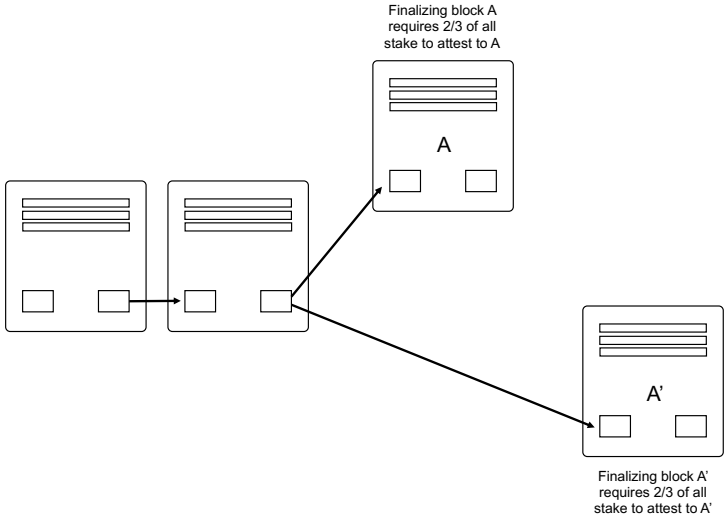
- Therefore, at least 1/3 of all stake signed both A and A'

Ethereum PoS: Punishing a Double-Spend Attacker



- Therefore, at least 1/3 of all stake signed both A and A'
- This stake that signed conflicting transactions is algorithmically destroyed ('slashed')

Ethereum PoS: Punishing a Double-Spend Attacker



- Therefore, at least 1/3 of all stake signed both A and A'
- This stake that signed conflicting transactions is algorithmically destroyed ('slashed')
- The reporter of the conflict earns a small bounty

Does Slashing Work?

- Recent work joint with Andrew Lewis-Pye and Tim Roughgarden.
- Our formal question: is there a permissionless consensus protocol that is "expensive to attack in the absence of collapse" (EAAC)?
- EAAC definition (informal): attacker loses their capital but honest participants are not harmed.

Does Slashing Work?

- Recent work joint with Andrew Lewis-Pye and Tim Roughgarden.
- Our formal question: is there a permissionless consensus protocol that is "expensive to attack in the absence of collapse" (EAAC)?
- EAAC definition (informal): attacker loses their capital but honest participants are not harmed.
- Our answers:

Does Slashing Work?

- Recent work joint with Andrew Lewis-Pye and Tim Roughgarden.
- Our formal question: is there a permissionless consensus protocol that is "expensive to attack in the absence of collapse" (EAAC)?
- EAAC definition (informal): attacker loses their capital but honest participants are not harmed.
- Our answers:
  - No: if the consensus protocol is "fully permissionless" or "dynamically available" (even in the synchronous communications model)

Does Slashing Work?

- Recent work joint with Andrew Lewis-Pye and Tim Roughgarden.
- Our formal question: is there a permissionless consensus protocol that is "expensive to attack in the absence of collapse" (EAAC)?
- EAAC definition (informal): attacker loses their capital but honest participants are not harmed.
- Our answers:
  - No: if the consensus protocol is "fully permissionless" or "dynamically available" (even in the synchronous communications model)
  - No: in the partially synchronous communications model (even for permissioned protocols)

Does Slashing Work?

- Recent work joint with Andrew Lewis-Pye and Tim Roughgarden.
- Our formal question: is there a permissionless consensus protocol that is "expensive to attack in the absence of collapse" (EAAC)?
- EAAC definition (informal): attacker loses their capital but honest participants are not harmed.
- Our answers:
  - No: if the consensus protocol is "fully permissionless" or "dynamically available" (even in the synchronous communications model)
  - No: in the partially synchronous communications model (even for permissioned protocols)
  - Yes! In the synchronous communications model, for quasi-permissionless consensus protocols, if the attacker is bounded above by $\frac{2}{3}$ of the total

Does Slashing Work?

- Recent work joint with Andrew Lewis-Pye and Tim Roughgarden.
- Our formal question: is there a permissionless consensus protocol that is "expensive to attack in the absence of collapse" (EAAC)?
- EAAC definition (informal): attacker loses their capital but honest participants are not harmed.
- Our answers:
  - No: if the consensus protocol is "fully permissionless" or "dynamically available" (even in the synchronous communications model)
  - No: in the partially synchronous communications model (even for permissioned protocols)
  - Yes! In the synchronous communications model, for quasi-permissionless consensus protocols, if the attacker is bounded above by $\frac{2}{3}$ of the total
  - No again, if the attacker can be $> \frac{2}{3}$ (Tas et al, 2023)

Does Slashing Work?

- Recent work joint with Andrew Lewis-Pye and Tim Roughgarden.
- Our formal question: is there a permissionless consensus protocol that is "expensive to attack in the absence of collapse" (EAAC)?
- EAAC definition (informal): attacker loses their capital but honest participants are not harmed.
- Our answers:
  - No: if the consensus protocol is "fully permissionless" or "dynamically available" (even in the synchronous communications model)
  - No: in the partially synchronous communications model (even for permissioned protocols)
  - Yes! In the synchronous communications model, for quasi-permissionless consensus protocols, if the attacker is bounded above by $\frac{2}{3}$ of the total
  - No again, if the attacker can be $> \frac{2}{3}$ (Tas et al, 2023)
- Translation:
  - Ethereum slashing works up to a $2/3$ attacker under assumptions about network reliability. Above $2/3$, then some external source of security is needed (e.g., law).
  - Consensus protocols like Bitcoin's cannot be EAAC (they are "fully permissionless")