RepuCoin: Your reputation is your power

Jiangshan Yu
Cybersecurity Lab, Monash University, Australia

Joint work with David Kozhaya (*), Jérémie Decouchant (†), and Paulo Esteves-Veríssimo (†)

(*) ABB Corporate Research Center, Switzerland
(†) SnT, University of Luxembourg, Luxembourg

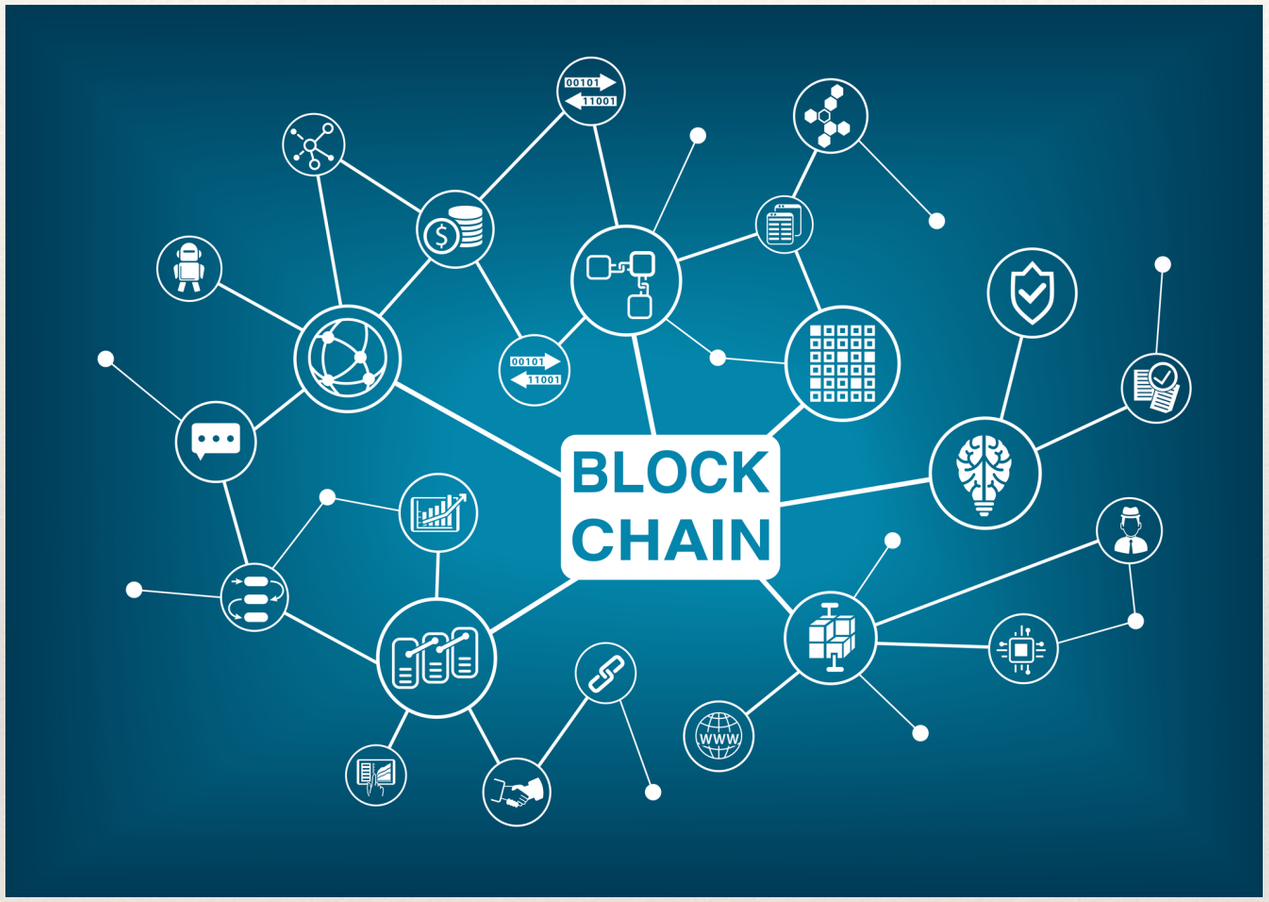
MONASH University

What is Blockchain?

A distributed database records all activities as transactions

MONASH University

Why Blockchain?

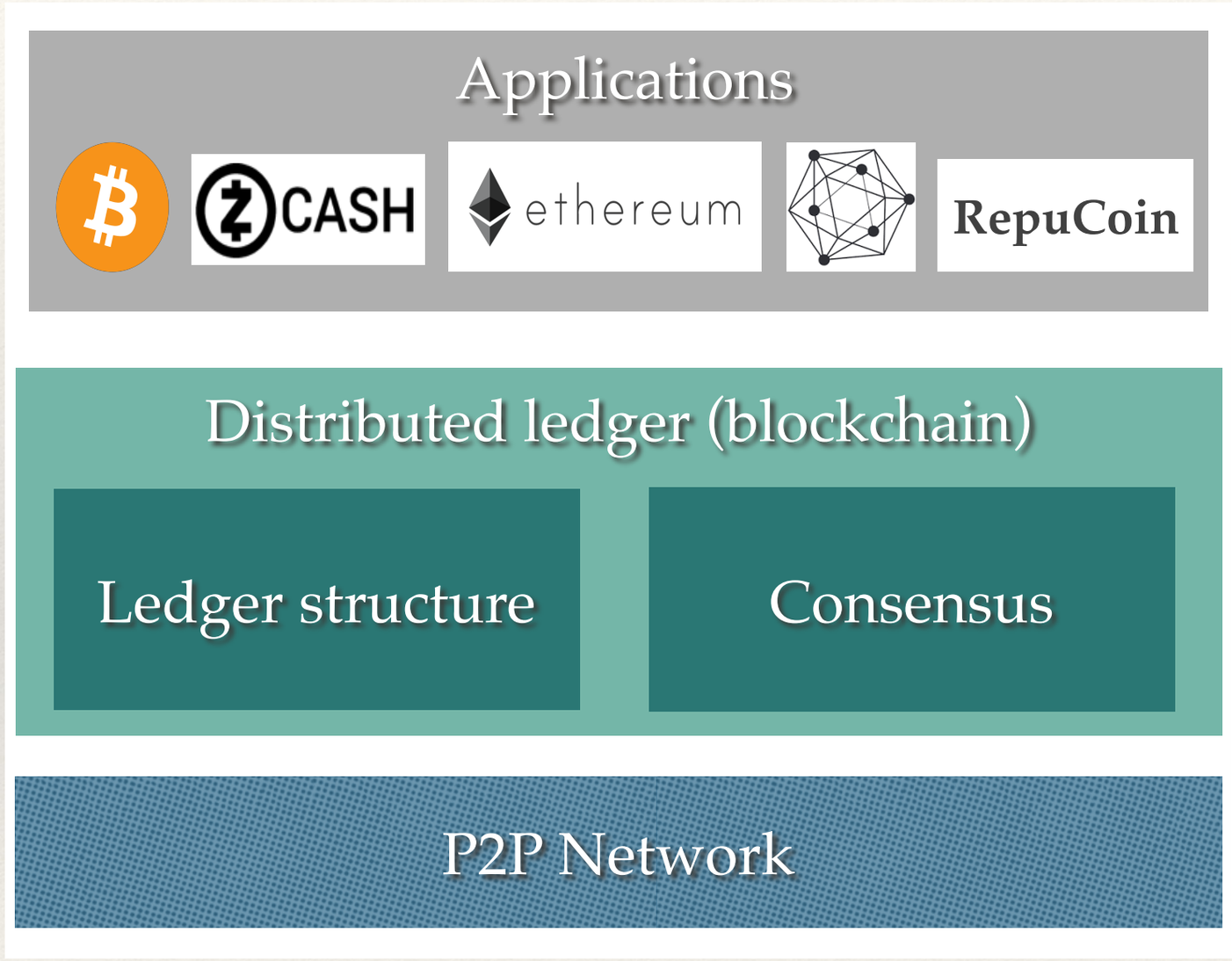


Goals:

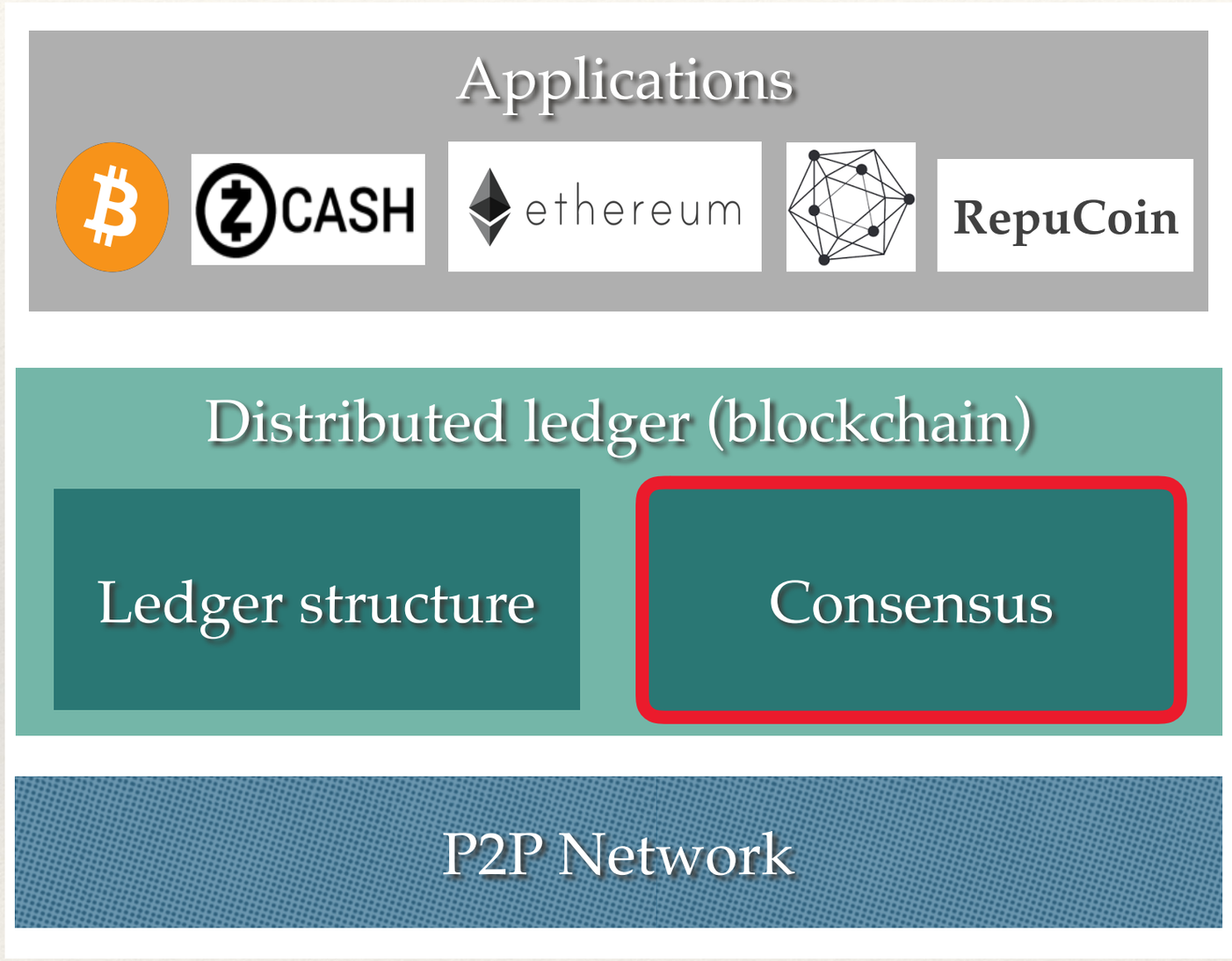
- Secure: non-changeable history
- Robust: no single point of failure
- Transparent: everyone can read
- TTP free: everyone can write

MONASH University

Architecture

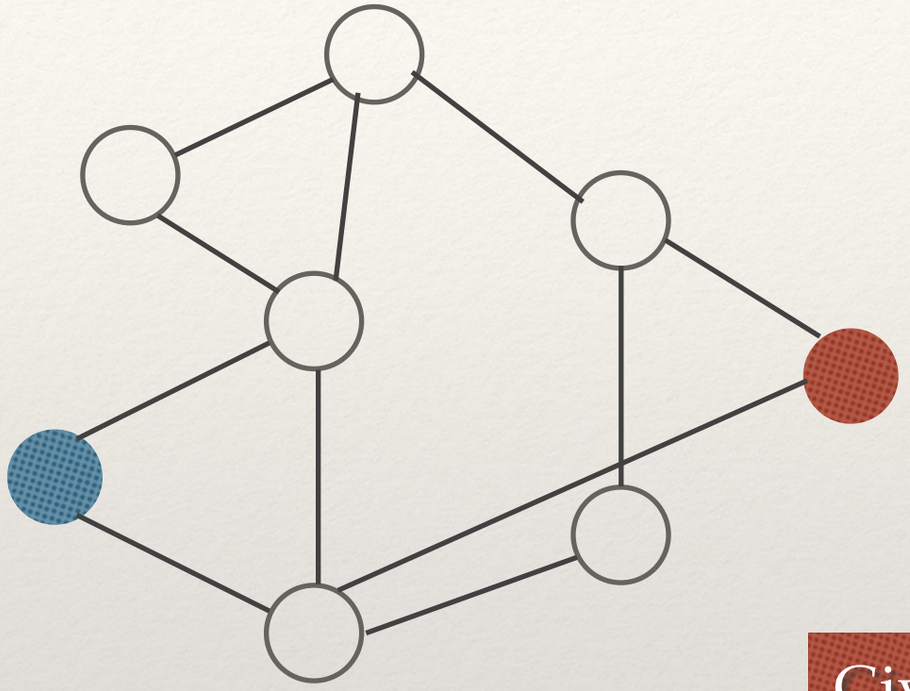


Architecture



Conflict transactions

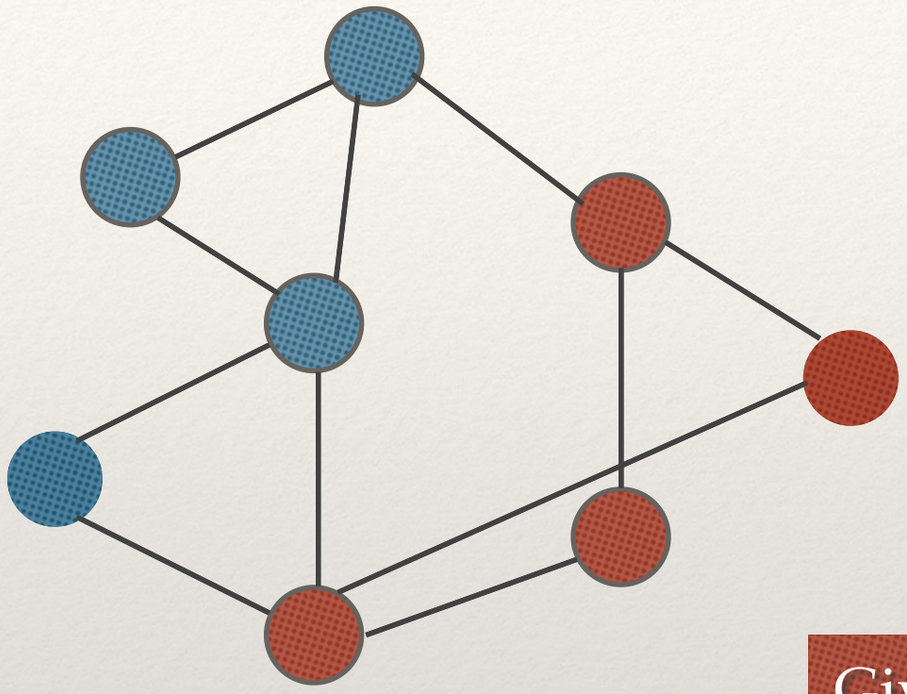
Give my coin **c_1** to Bob



Give my coin **c_1** to Chris

Conflict transactions

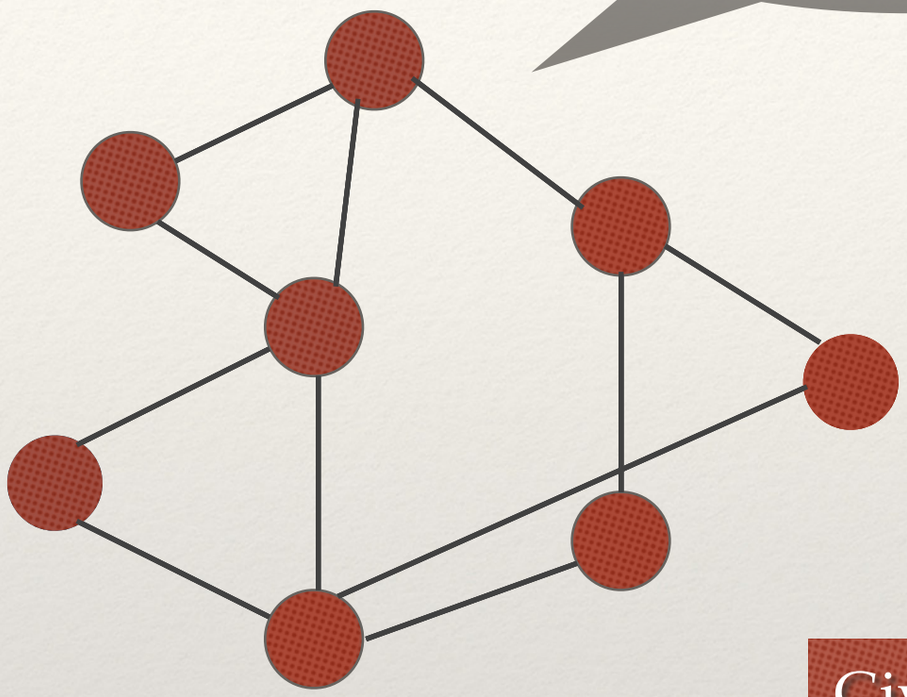
Give my coin **c_1** to Bob



Give my coin **c_1** to Chris

Conflict transactions

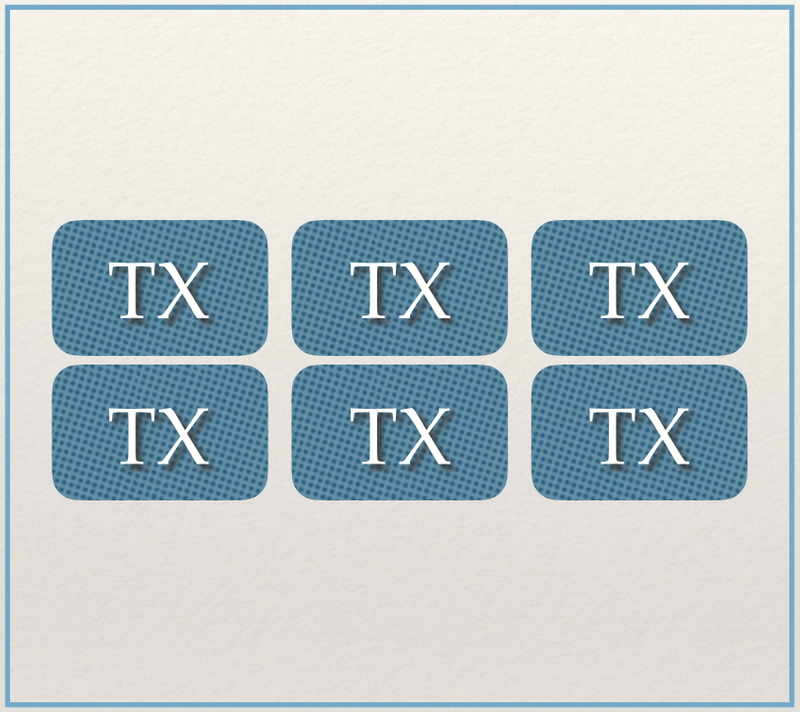
Give my coin **c_1** to Bob



Give my coin **c_1** to Chris

MONASH University

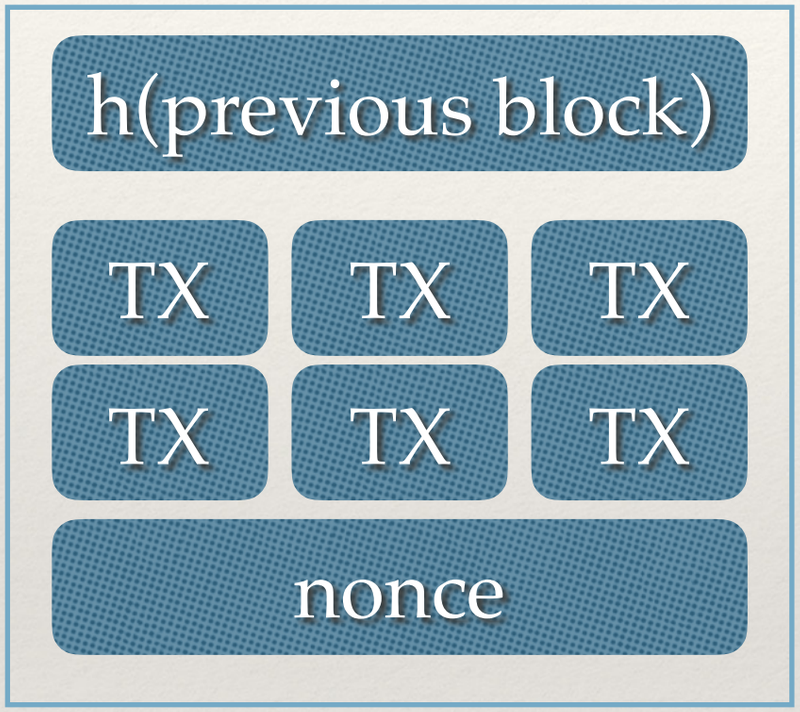
Bitcoin: Proof of work



A block

$$\begin{array}{c}\text{for nonce in } range(0, 2^{32}):\\ \text{if h(block)} < \text{target:}\\ \text{print (`success')}\\ \text{print (nonce)}\end{array}$$

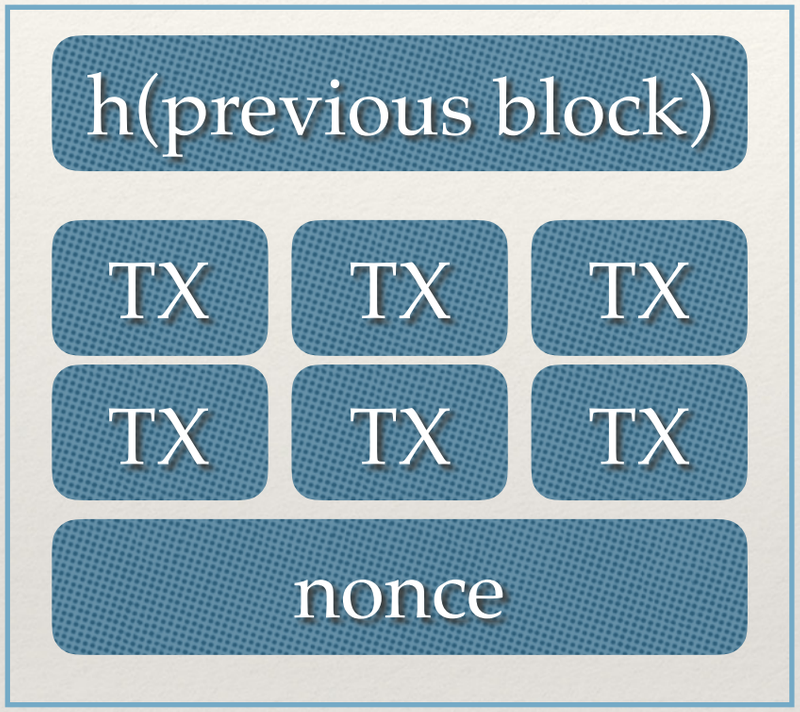
Bitcoin: Proof of work



A block

$$\begin{array}{c}\text{for nonce in } range(0, 2^{32}):\\ \text{if h(block)} < \text{target:}\\ \text{print (`success')}\\ \text{print (nonce)}\end{array}$$

Bitcoin: Proof of work



A block

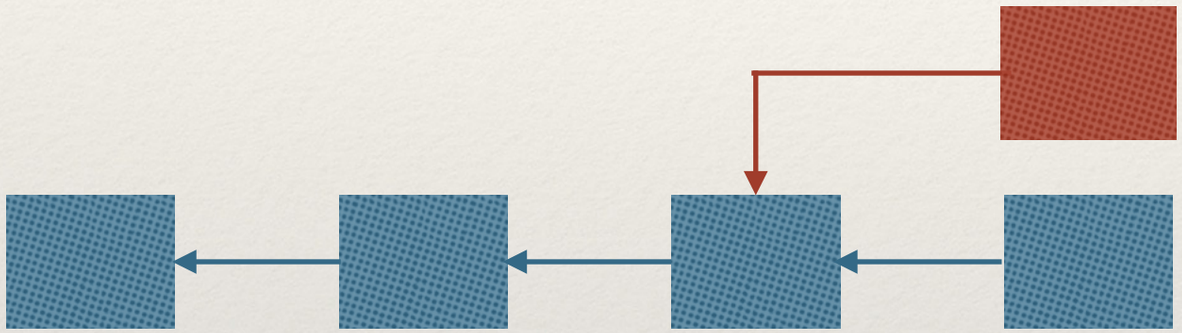
$$\begin{array}{c}\text{for nonce in } range(0, 2^{32}):\\ \text{if h(block)} < \text{target:}\\ \text{print (`success')}\\ \text{print (nonce)}\end{array}$$

Problem A: **Slow TX validation**

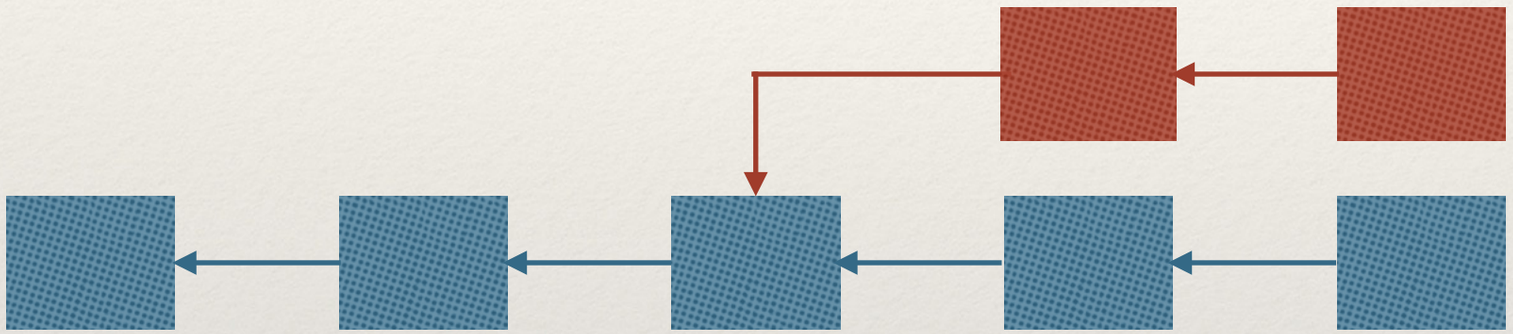
10 mins/block, 7 transactions per second (TPS)

Problem B: **multiple valid solutions**

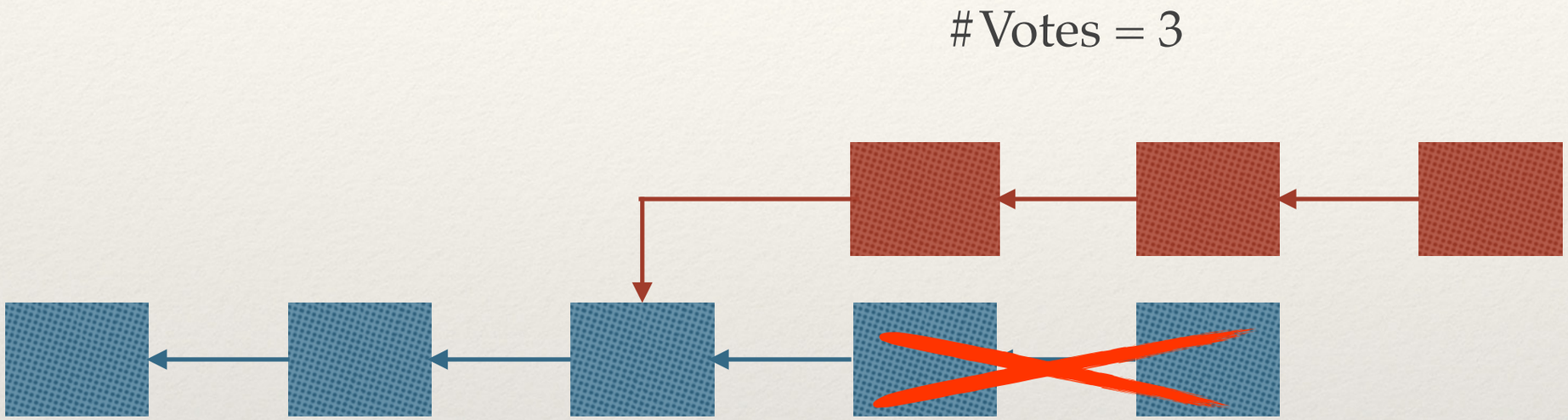
Blockchain: resolving forks



Blockchain: resolving forks



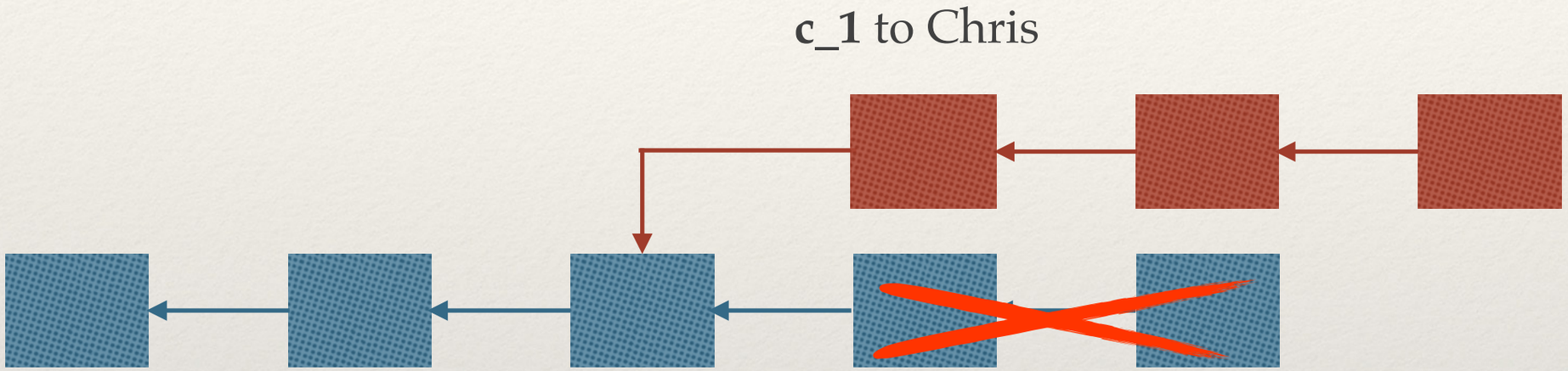
Blockchain: resolving forks



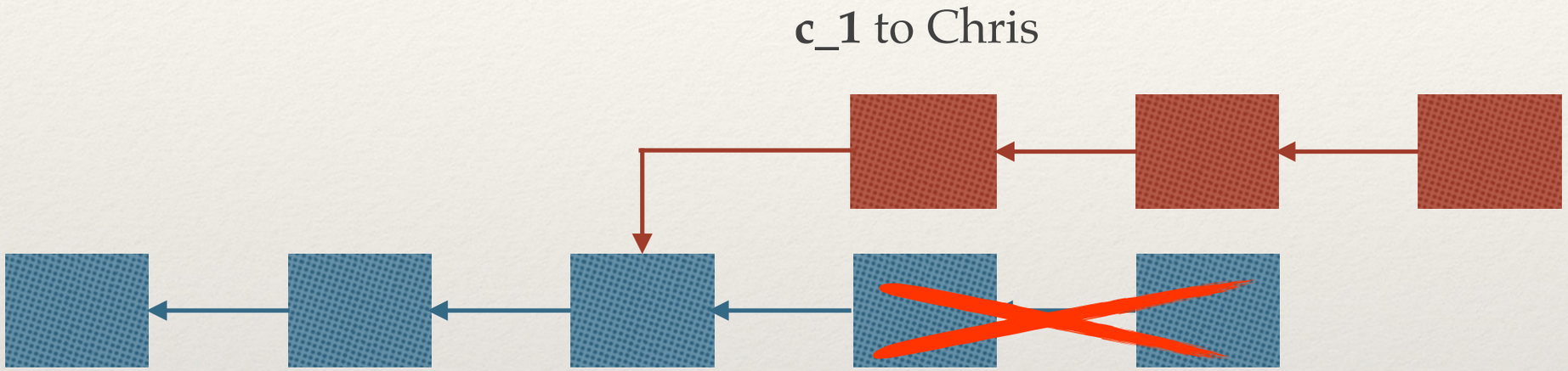
Double spending attack

c_1 to Bob

Double spending attack

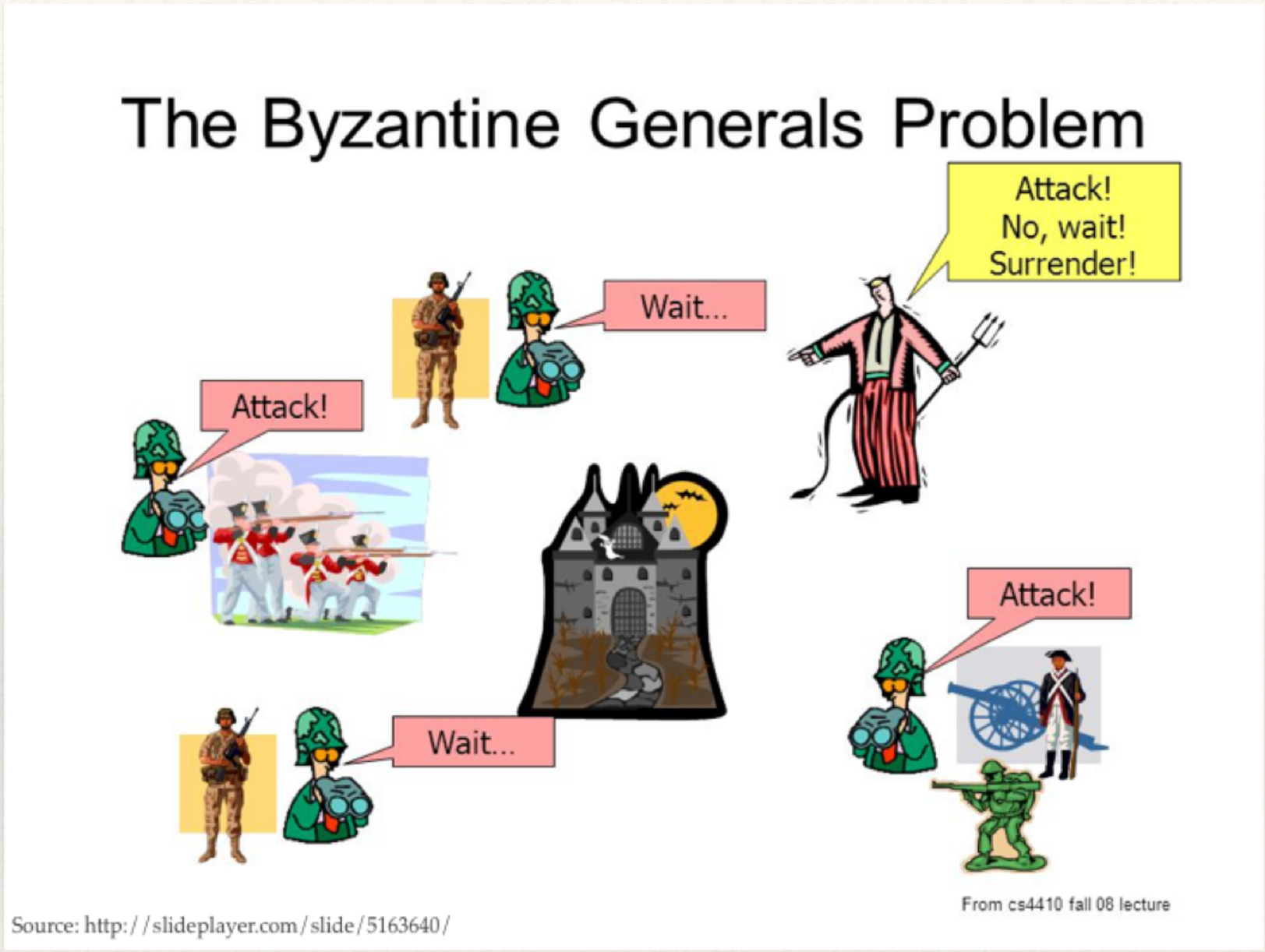


Double spending attack



If an attacker has >50% CPU power, it can spend a coin more than once.

Dependability: 40 years of BFT research



Lamport, L.; Shostak, R.; Pease, M. (1982). "The Byzantine Generals Problem". ACM Trans. on Programming Languages and Systems. 4 (3): 382–401

Mind the gap

	BFT protocols	Permissionless Blockchain
Openess	A pre-fixed committee for voting	Open to everyone
Non-malicious participants	Honest	Honest or rational
Assumption	Liveness $f \leq \left\lfloor \frac{n-1}{3} \right\rfloor$	At least one active miner
	Safety $f \leq \left\lfloor \frac{2(n-1)}{3} \right\rfloor$	f < 50% mining power (BTC)
# voters	Small (n in total)	Large
# participants	n in total; f malicious	??

Applying BFT to Blockchain

Permissioned (consortium) Blockchain

A good start, but not the end…

BFT and Permissionless Blockchain

Challenge for system deployment:
How to define n? And hence predict f?

- *n is dynamic and can become very large*
- *In practice, in an open BFT-based system, we cannot guarantee that an attacker will not control more than a priori defined f nodes*

BFT and Permissionless Blockchain

Several prior efforts on applying BFT to Blockchain

- PeerCensus
- ByzCoin
- Solida
- Hybrid consensus
- Thunderella
- …

PoW+BFT

- Setp 1. Run PoW to select a small number of members;
- Setp 2. Run BFT to reach agreement

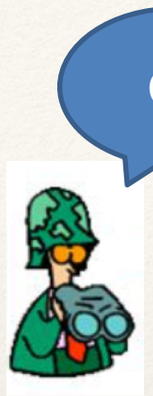
n could be fixed and small this way

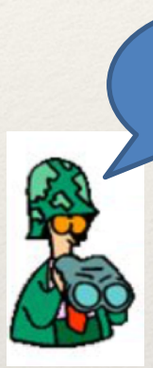
So, we could predict f …

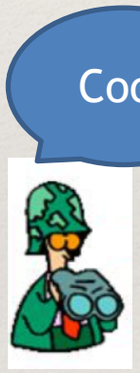
… *Could we?* ...

Assumption v.s. Reality

Byzantine generals plan!



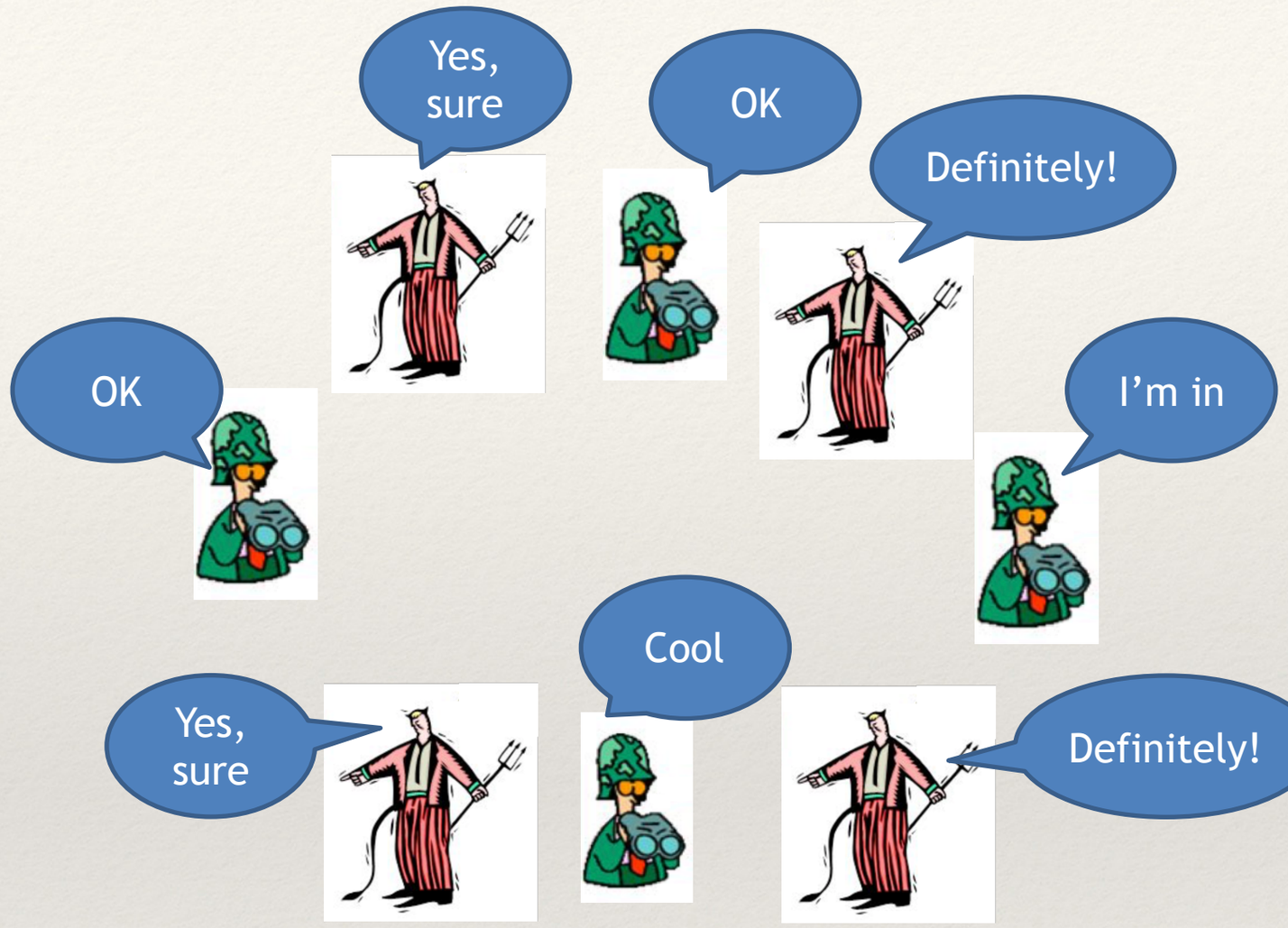




Reality is….
If anyone can be selected to run consensus,
how can we be sure that the system contains no more than f malicious nodes?

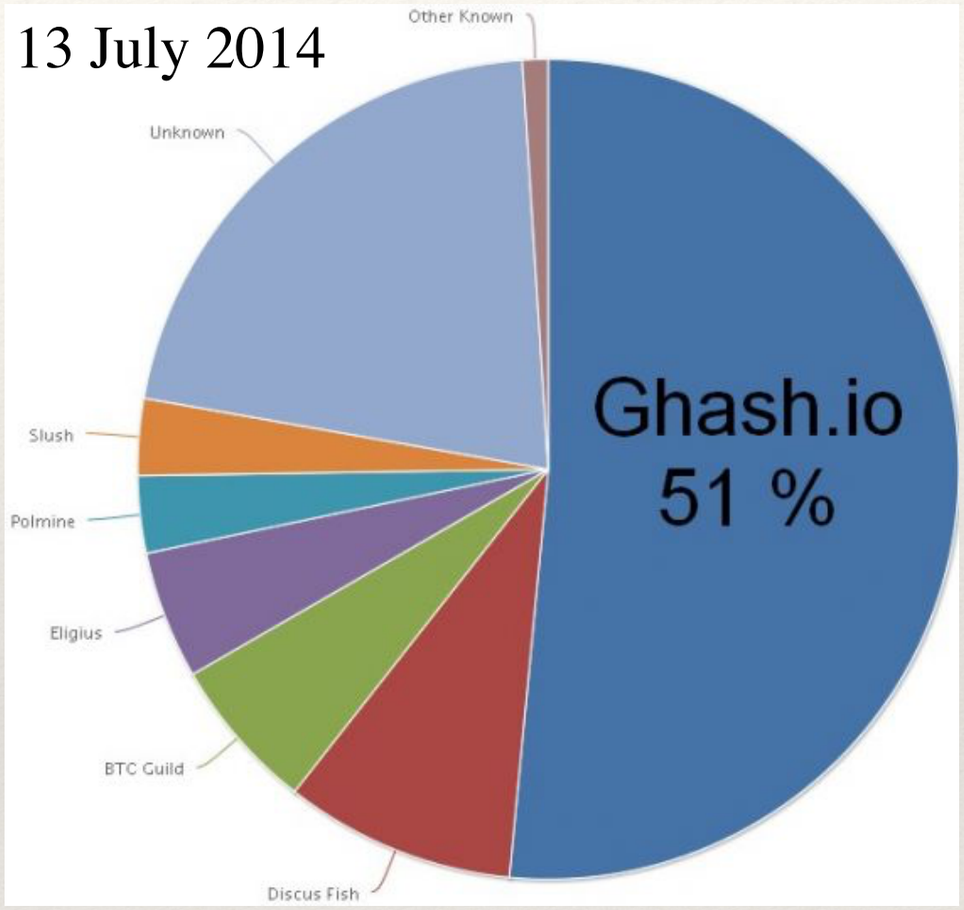
Assumption v.s. Reality

Byzantine generals plan!



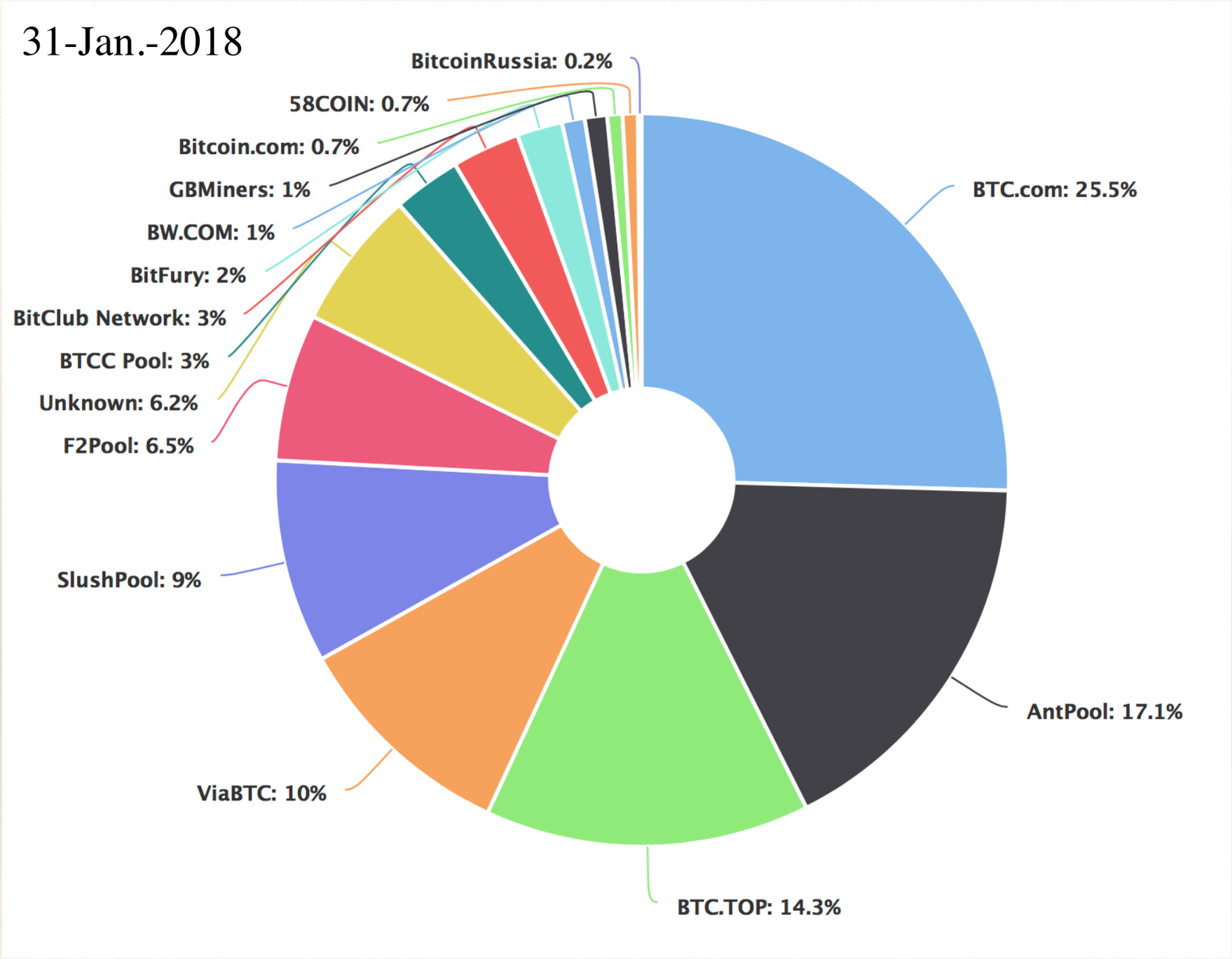
Reality is….
If anyone can be selected to run consensus,
how can we be sure that the system contains no more than f malicious nodes?

Reality is tough



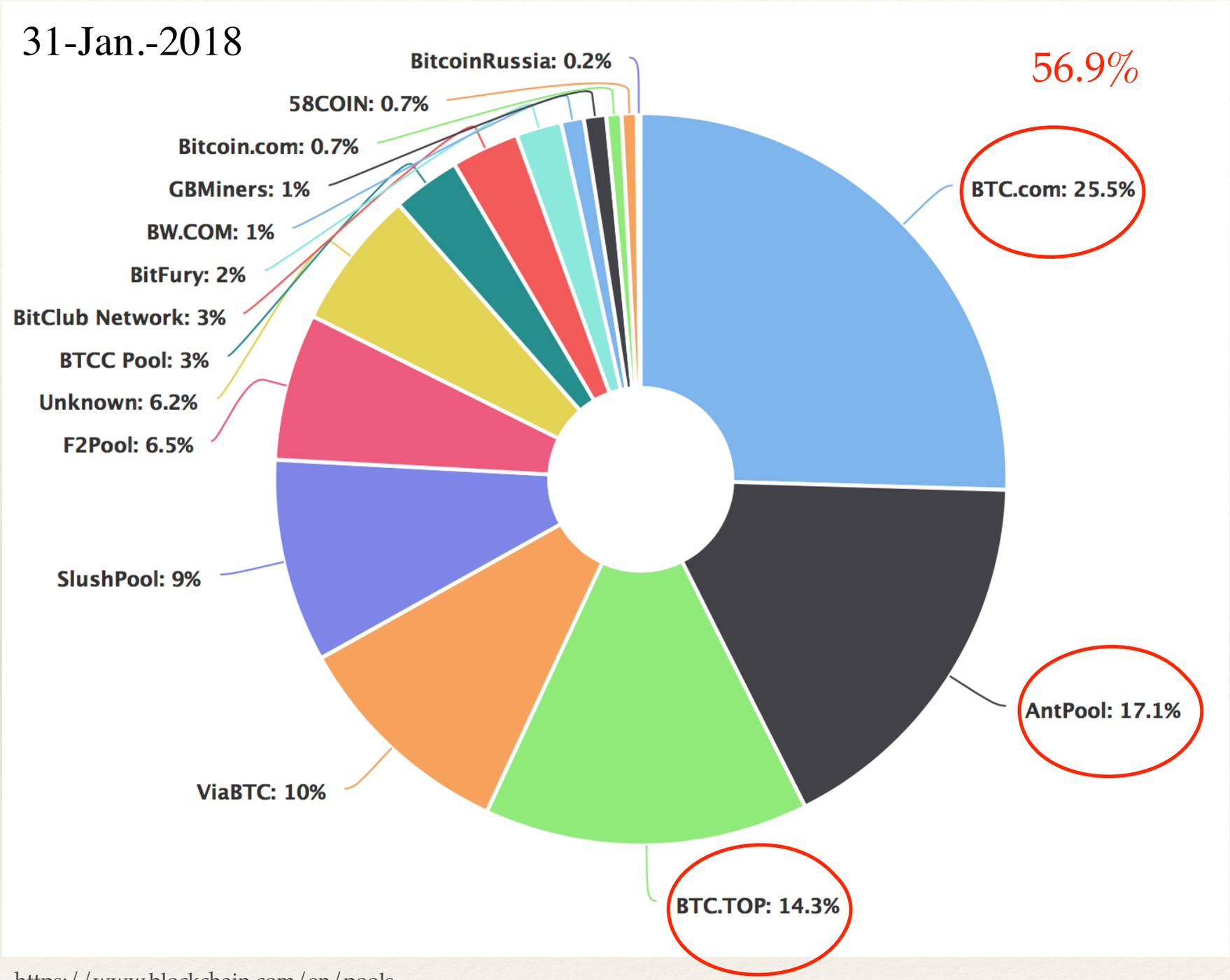
MONASH University

Jan. 2018



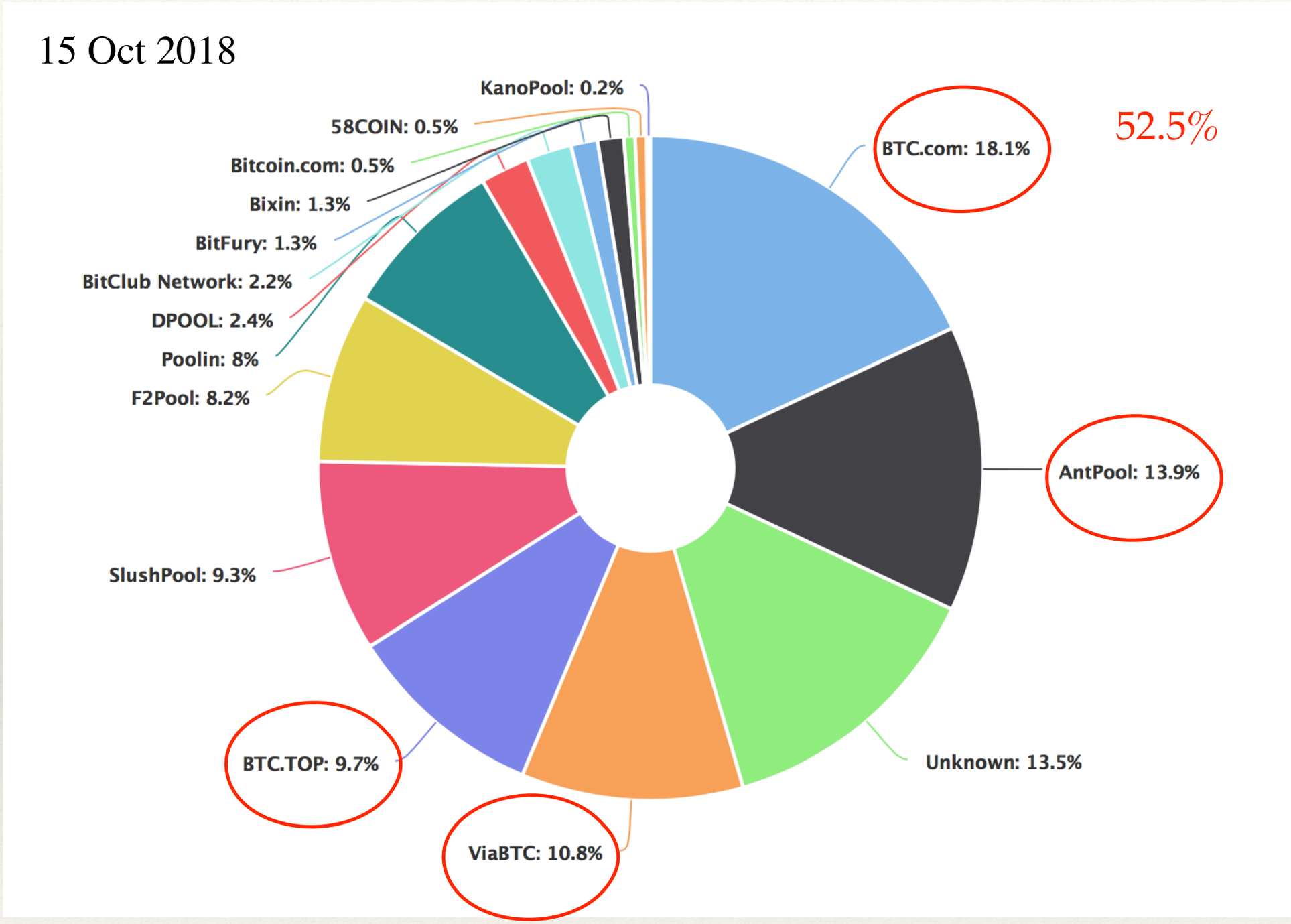
https://www.blockchain.com/en/pools

Jan. 2018



https://www.blockchain.com/en/pools

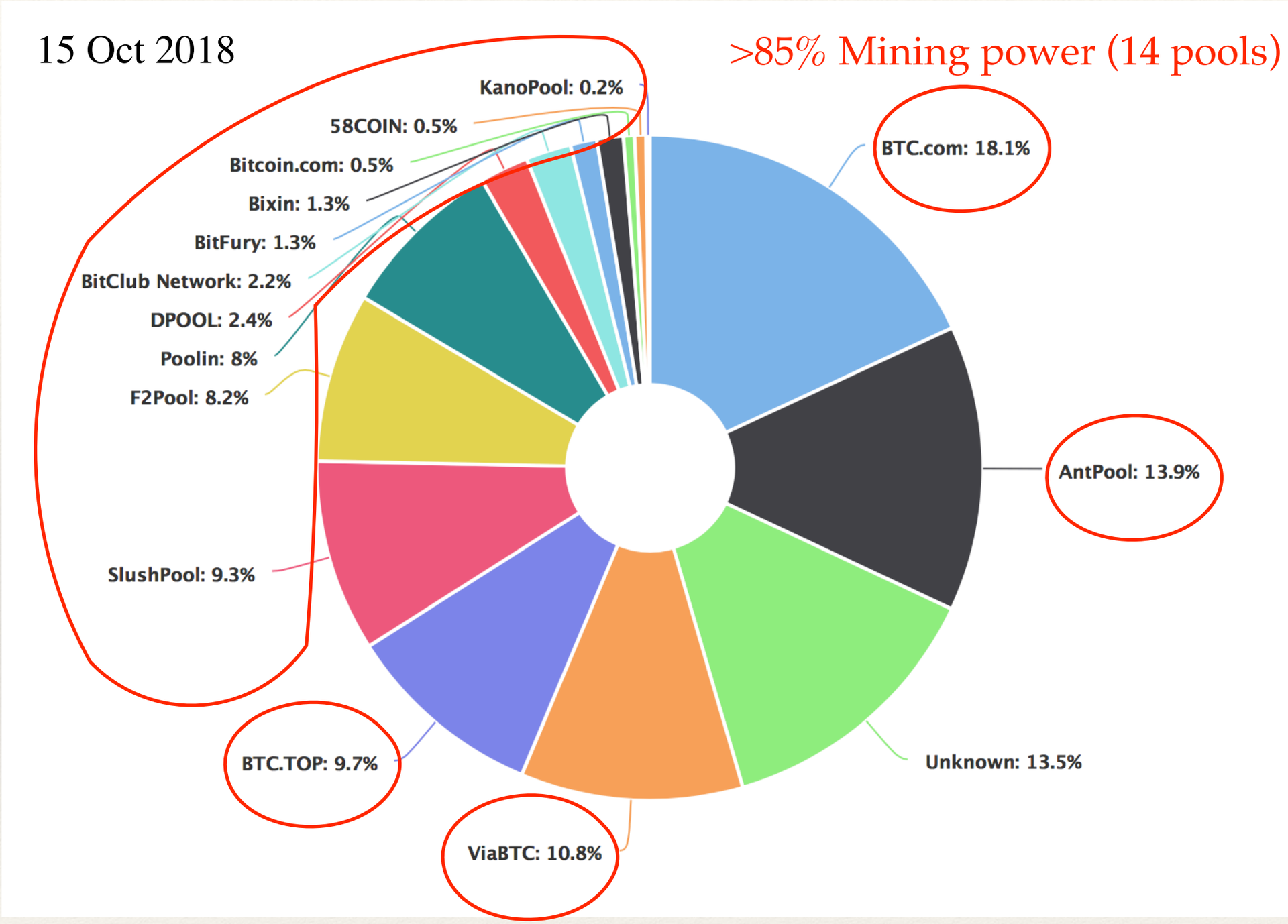
Oct. 2018



https://www.blockchain.com/en/pools

MONASH University

Oct. 2018



https://www.blockchain.com/en/pools

MONASH University

Reality is tough

2013

Majority is not Enough: Bitcoin Mining is Vulnerable*

Ittay Eyal and Emin Gün Sirer

Department of Computer Science, Cornell University
ittay.eyal@cornell.edu, egs@systems.cs.cornell.edu

Abstract. The Bitcoin cryptoc[illegible]its transactions in a public log called the blockchain. Its s[illegible]tically on the distributed protocol that maintains the blockchain, run by participants called miners. Conventional wisdom asserts that the mining protocol is incentive-compatible and secure against colluding minority groups, that is, it incentivizes miners to follow the protocol as prescribed.

We show that the Bitcoin mining protocol is not incentive-compatible. We present an attack with which colluding miners obtain a revenue larger than their fair share. This attack can have significant consequences for Bitcoin: Rational miners will prefer to join the selfish miners, and the colluding group will increase in size until it becomes a majority. At this point, the Bitcoin system ceases to be a decentralized currency.

> 25%

MONASH University

Reality is tough

2016

Why buy when you can rent?
Bribery attacks on Bitcoin-style consensus

Joseph Bonneau

Stanford University & Electronic Frontier Foundation

Abstract. The Bitcoin cryptocurrency introduced a novel distributed consensus mechanism relying on economic incentives. While a coalition controlling a majority of computational power may undermine the system, for example by double-spending funds, it is often assumed it would be incentivized not to attack to protect its long-term stake in the health

>50% CPU power for a short time.
(flash attack)

All existing PoW-based systems are vulnerable to this attack.

public, distributed ledger called the blockchain which logs all transactions to ensure that funds may only be spent once. Bitcoin uses a computational puzzle

The big big challenge

In a permissionless blockchain, how to enforce, at least with a very high probability, that

malicious_nodes ≤ f?

ΣP malicious_nodes $\leq P_f$?

RepuCoin Overview

Main problems of PoW:	Our solutions:
Decision (voting) power is CPU power - **Instantaneous** power - can be gained **quickly**; - vulnerable to flash attacks.	Decision (voting) power is reputation - **Integrated** power (past performance) - can only grow **slowly with bounded speed**; - **Not** vulnerable to flash attacks.

RepuCoin Overview

Main problems of PoW:	Our solutions:
Decision (voting) power is CPU power - **Instantaneous** power - can be gained **quickly**; - vulnerable to flash attacks.	Decision (voting) power is reputation - **Integrated** power (past performance) - can only grow **slowly with bounded speed**; - **Not** vulnerable to flash attacks.
Rationality and maliciousness - not clearly distinguished	**Rationality and maliciousness** - separate protection measures

RepuCoin Overview

Main problems of PoW:	Our solutions:
Decision (voting) power is CPU power - **Instantaneous** power - can be gained **quickly**; - vulnerable to flash attacks.	Decision (voting) power is reputation - **Integrated** power (past performance) - can only grow **slowly with bounded speed**; - **Not** vulnerable to flash attacks.
Rationality and maliciousness - not clearly distinguished	**Rationality and maliciousness** - separate protection measures
PoW consensus is **probabilistic** - forkable BC	**PoR** consensus is **deterministic** - novel weighted voting consensus algorithm - non-forkable BC

RepuCoin Overview

Main problems of PoW:	Our solutions:
Decision (voting) power is CPU power - **Instantaneous** power - can be gained **quickly**; - vulnerable to flash attacks.	Decision (voting) power is reputation - **Integrated** power (past performance) - can only grow **slowly with bounded speed**; - **Not** vulnerable to flash attacks.
Rationality and maliciousness - not clearly distinguished	**Rationality and maliciousness** - separate protection measures
PoW consensus is **probabilistic** - forkable BC	**PoR** consensus is **deterministic** - novel weighted voting consensus algorithm - non-forkable BC
Low (stochastic) resilience - vulnerable to selfish mining (>25%) and other attacks leveraging instantaneous power	**High** (stochastic) resilience -**Not** vulnerable to instantan. power attacks -**Non-rationality** of infiltration attacks

RepuCoin Overview

Main problems of PoW:	Our solutions:
Decision (voting) power is CPU power - **Instantaneous** power - can be gained **quickly**; - vulnerable to flash attacks.	Decision (voting) power is reputation - **Integrated** power (past performance) - can only grow **slowly with bounded speed**; - **Not** vulnerable to flash attacks.
Rationality and maliciousness - not clearly distinguished **PoW** consensus is **probabilistic** - forkable BC	**Rationality and maliciousness** - separate protection measures **PoR** consensus is **deterministic** - novel weighted voting consensus algorithm - non-forkable BC
Low (stochastic) resilience - vulnerable to selfish mining (>25%) and other attacks leveraging instantaneous power	**High** (stochastic) resilience -**Not** vulnerable to instantan. power attacks -**Non-rationality** of infiltration attacks
Low Throughput: - 7 TPS - 1,000 TPS (ByzCoin)	**High** Throughput: - (fast) PoR for committing transactions - 10,000 TPS (256 Byte per TX)

The logic of RepuCoin in a nutshell

- reputation-based weighted voting consensus is safe and live as long as relative decision power (given by reputation score) of attackers is below a defined threshold, fraction of the total

- max rate of decision power growth of any system participant is deterministic, bounded and known, imposed by the proof-of-reputation function

- there is no rational economic model for infiltration attacks --- compared to the cost of attacking different systems

- attacks attacks on liveness or safety still being possible, the network achieves very high stochastic robustness against them --- i.e., attack effort to reach network control compares very favorably to previous works

- RepuCoin prevents all currently known attacks.

How does RepuCoin Work?

1. Miners gain reputation slowly with a bounded rate by contributing to the blockchain

Algorithm 2 Reputation algorithm

Input: L, $\{k_i\}_{i=1}^{t}$, $\{m_j\}_{j=1}^{N_l}$, m, c, a, and λ.
Output: Reputation $R \in [0, 1]$ of the corresponding miner.

1: $\text{mean}_k = \frac{\sum_{i=1}^{t} k_i}{L}$
2: $\text{mean}_m = \frac{1}{N_l} \cdot \sum_{j=1}^{N_l} \frac{m_j}{m}$
3: $s_k = \sqrt{\frac{1}{t} \cdot \sum_{i=1}^{t} (\frac{k_i}{} - \frac{\sum_{i=1}^{t} k_i}{})^2}$
4: $s_m = \sqrt{\frac{1}{N_l} \cdot \sum_{j=}}$
5: $y_1 = \frac{\text{mean}_k}{1+s_k}$
6: **if** $N_l \geq 1$ **then**
$y_2 = \frac{\text{mean}_m}{1+s_m}$
7: **else**
8: $y_2 = 1$
9: **end if**
10: $x = y_1 \cdot y_2 \cdot L$
11: $f(x) = \frac{1}{2}(1 + \frac{x-a}{\lambda+\lvert x-a \rvert})$
12: $R = min(1, H \cdot (Ext + f(x)))$

- total amount of valid work
- regularity of that work

How does RepuCoin Work?

1. Miners gain reputation slowly with a bounded rate by contributing to the blockchain

2. Top reputed miners dynamically form a consensus committee

How does RepuCoin Work?

1. Miners gain reputation slowly with a bounded rate by contributing to the blockchain

2. Top reputed miners dynamically form a consensus committee

3. The committee votes through reputation-based weighted voting protocol to “pin” keyblocks;

MONASH University

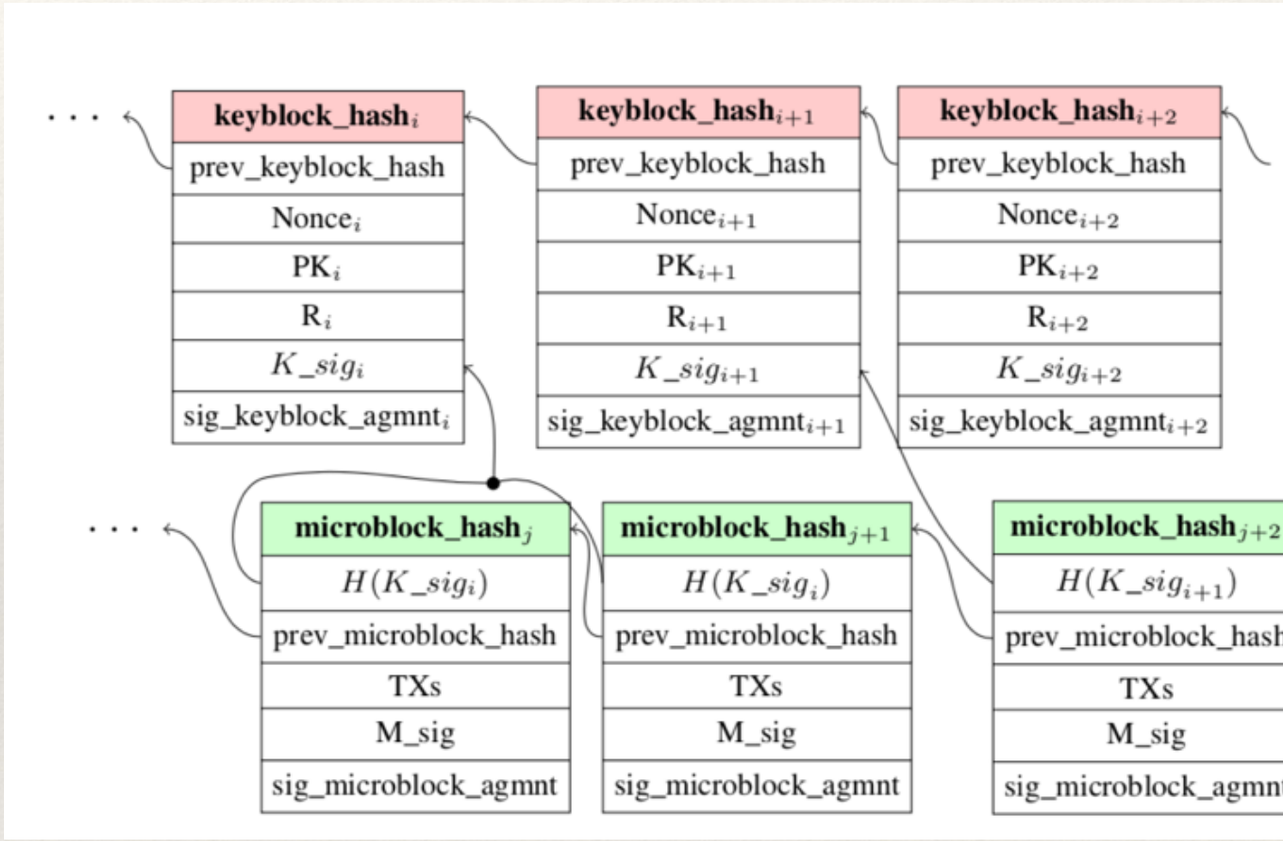
How does RepuCoin Work?

1. Miners gain reputation slowly with a bounded rate by contributing to the blockchain

2. Top reputed miners dynamically form a consensus committee

3. The committee votes through reputation-based weighted voting protocol to “pin” keyblocks;

4. A randomly elected leader proposes microblocks to the committee for their approval;



MONASH University

How does RepuCoin Work?

1. Miners gain reputation slowly with a bounded rate by contributing to the blockchain

2. Top reputed miners dynamically form a consensus committee

3. The committee votes through reputation-based weighted voting protocol to “pin” keyblocks;

4. A randomly elected leader proposes microblocks to the committee for their approval;

5. Mis-behaved miners will be punished, and they lose reputation

How does RepuCoin Work?

- We have “Keyblocks” and “Microblocks”
  - Miners solve Bitcoin-like puzzles to create keyblocks (Keyblocks do not contain any transaction)

$$H(prev\_keyblock\_hash||Nonce||PK) < target$$

  - Microblocks contain transactions, created by a random leader

- Each block is “pinned” by the consensus group

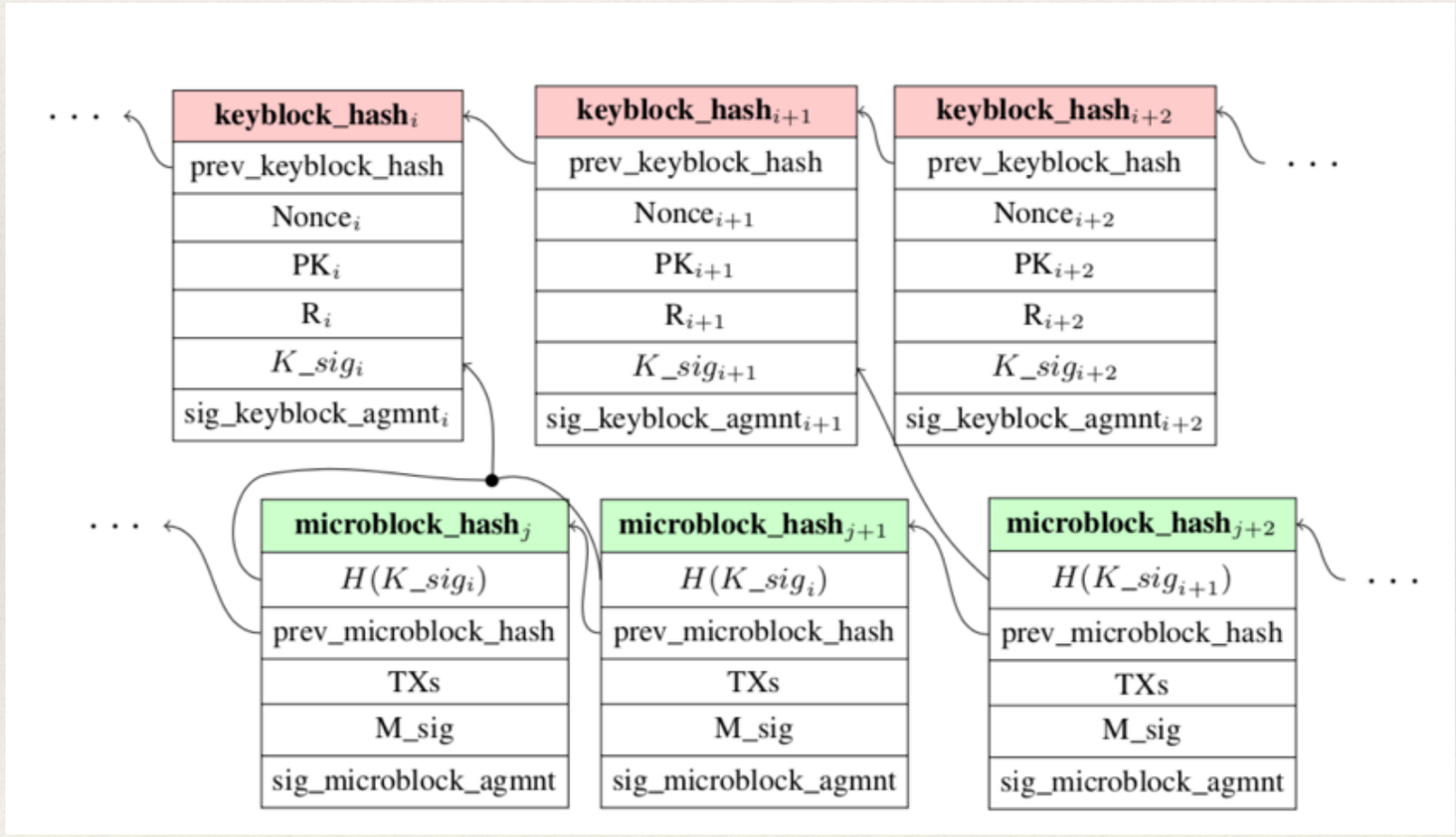
Block pinning

- Top reputed miners dynamically form a consensus group
- Each keyblock is proposed to the consensus group to be “pinned” through reputation-based weighted voting consensus
- A new leader is randomly determined by the newly pinned keyblock

$$l_i := x_j \quad s.t.\ x_j \in \mathbb{X} \ \wedge \ j = H(K\_sig_i) \mod |\mathbb{X}|$$

Block pinning

- The elected leader commits transactions into microblocks
- Microblocks are also pinned by the consensus group
- Microblocks are linked back to the keyblock



MONASH University

Reputation is your power

Control over reputation

1. **Gain reputation** by contributing ‘good’ work
    1. total amount of contributed,
    2. the regularity of that work
    3. calculated over the contribution of the entire system of all periods
2. **Lose reputation** if misbehaved

Algorithm 2 Reputation algorithm

Input: L, $\{k_i\}_{i=1}^{t}$, $\{m_j\}_{j=1}^{N_l}$, m, c, a, λ, H, and Ext.
Output: Reputation $R \in [0, 1]$ of the corresponding miner.

1: $\text{mean}_k = \frac{\sum_{i=1}^{t} k_i}{L}$
2: $\text{mean}_m = \frac{1}{N_l} \cdot \sum_{j=1}^{N_l} \frac{m_j}{m}$
3: $s_k = \sqrt{\frac{1}{t} \cdot \sum_{i=1}^{t} (\frac{k_i}{c} - \frac{\sum_{i=1}^{t} k_i}{L})^2}$
4: $s_m = \sqrt{\frac{1}{N_l} \cdot \sum_{j=1}^{N_l} (\frac{m_j}{m} - \frac{1}{N_l} \cdot \sum_{j=1}^{N_l} \frac{m_j}{m})^2}$
5: $y_1 = \frac{\text{mean}_k}{1+s_k}$
6: **if** $N_l \geq 1$ **then**
$y_2 = \frac{\text{mean}_m}{1+s_m}$
7: **else**
8: $y_2 = 1$
9: **end if**
10: $x = y_1 \cdot y_2 \cdot L$
11: $f(x) = \frac{1}{2}(1 + \frac{x-a}{\lambda+|x-a|})$
12: $R = min(1, H \cdot (Ext + f(x)))$

Reputation is your power

Control over reputation

1. **Gain reputation** by contributing ‘good’ work
    1. total amount of contributed,
    2. the regularity of that work
    3. calculated over the contribution of the entire system of all periods
2. **Lose reputation** if misbehaved

Algorithm 2 Reputation algorithm

Input: L, $\{k_i\}_{i=1}^{t}$, $\{m_j\}_{j=1}^{N_l}$, m, c, a, λ, H, and Ext.
Output: Reputation $R \in [0, 1]$ of the corresponding miner.

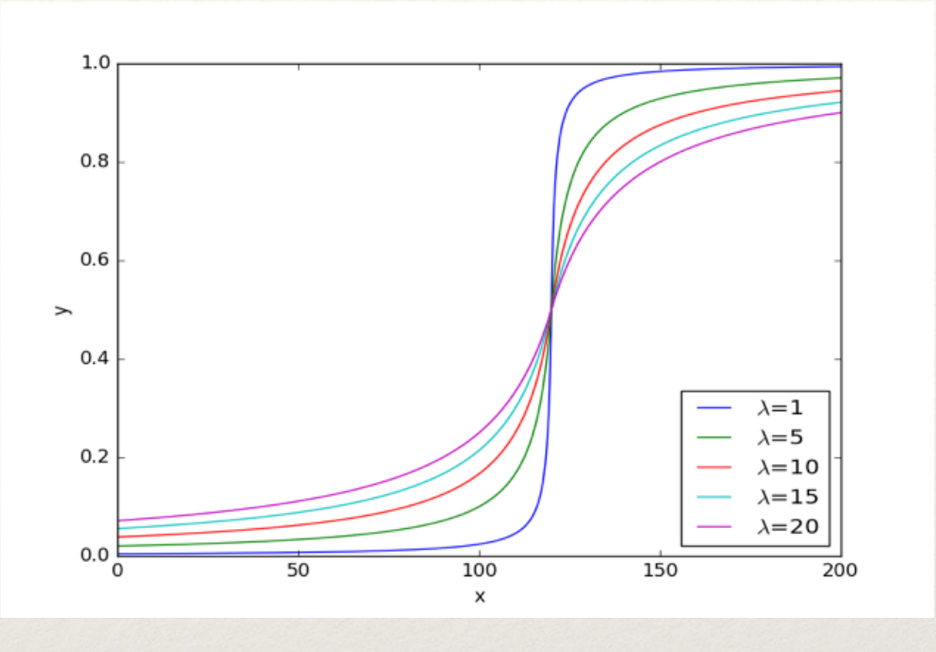
1: $\text{mean}_k = \frac{\sum_{i=1}^{t} k_i}{L}$
2: $\text{mean}_m = \frac{1}{N_l} \cdot \sum_{j=1}^{N_l} \frac{m_j}{m}$
3: $s_k = \sqrt{\frac{1}{t} \cdot \sum_{i=1}^{t} (\frac{k_i}{c} - \frac{\sum_{i=1}^{t} k_i}{L})^2}$
4: $s_m = \sqrt{\frac{1}{N_l} \cdot \sum_{j=1}^{N_l} (\frac{m_j}{m} - \frac{1}{N_l} \cdot \sum_{j=1}^{N_l} \frac{m_j}{m})^2}$
5: $y_1 = \frac{\text{mean}_k}{1+s_k}$
6: **if** $N_l \geq 1$ **then**
$y_2 = \frac{\text{mean}_m}{1+s_m}$
7: **else**
8: $y_2 = 1$
9: **end if**
10: $x = y_1 \cdot y_2 \cdot L$
11: $f(x) = \frac{1}{2}(1 + \frac{x-a}{\lambda+|x-a|})$
12: $R = min(1, H \cdot (Ext + f(x)))$

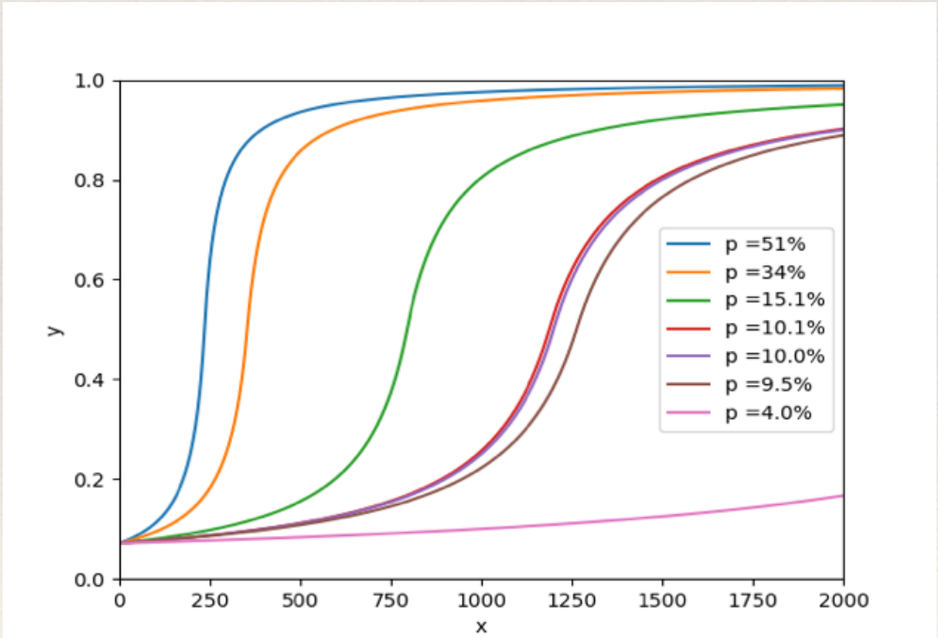
$R = min(1, H \cdot (Ext + f(x)))$ →

Helps bootstraping, and supports both permissioned and permissionless ledger

Reputation is your power

3. The **social objectives** of reputation:
    i. **careful start**, through an initial slow increase;
    ii. potential for quick reward of mature participants, through **fast increase in mid-life**;
    iii. prevention of over-control, by **slow increase near the top**





MONASH University

Reputation is your power

Reputation distribution of miners over time.

Time	[0, 0.2)	[0.2, 0.4)	[0.4, 0.6)	[0.6, 0.8)	[0.8, 1]
1 month	100%	-	-	-	-
6 months	64.7%	35.3%	-	-	-
1 year	21.8%	78.2%	-	-	-
2 years	9.6%	31.7%	38.1%	15.2%	-
3 years	2.7%	21.6%	19.5%	38.1%	15.2%
4 years	2.7%	19.1%	-	25%	53.2%
4 years	2.7%	15.1%	4%	17.9%	60.3%
20 years	0.4%	2.3%	-	3%	94.3%

Reputation is your power

Reputation-based incentives lead miners to work diligently and honestly

A successful miner

1. gets all mining rewards
2. shares transation fees with a randomly selected leader, according to the reputation.
3. gets >60 times better transaction fees than BTC, due to high throughput

Algorithm 1 Reward sharing algorithm

Input: The sequence $\mathbb{M} = \{m_0, m_1, \ldots, m_{n-1}\}$ of microblocks pinned in the $(i-1)$-th epoch, the signature K_sig_i contained in the i-th pinned keyblock, and the reputation R of the miner who created the $(i-1)$-th keyblock.

Output: Two subsets $\mathbb{M}', \mathbb{M}'' \subseteq \mathbb{M}$ of microblocks, where transaction fees contained in $\mathbb{M}'$ (resp. $\mathbb{M}''$) are allocated to the miner (resp. the leader) as reward.

$i' = H(K\_sig_i) \mod n$
$k = 0$
$\mathbb{M}' = \emptyset$
while $k < R \cdot n$ **do**
    $j = i' + k \mod n$
    $\mathbb{M}' = \mathbb{M}' \cup \{m_j\}$
    $k = k + 1$
end while
$\mathbb{M}'' = \mathbb{M} \setminus \mathbb{M}'$

Reputation is your power

Voting power is *integrated power*

(i.e. the *past* work over time, toward reputation)

rather than *instantaneous power*

(i.e. the *now* sheer computing power)

- Only top reputed nodes get involved in the consensus;
- The voting power of a top reputed node is the ratio of its reputation to all top reputed nodes

The weight of x_i's vote is $\frac{R_i}{\sum_{i=1}^{|\mathbb{X}|} R_i}$, for all possible x_i.

What do we enforce?

The increase of any miner's voting power is bounded by "physics"!

$$\frac{d^2Pd}{dN \cdot dt} = \frac{1}{2} \frac{\lambda}{(\lambda + |x - a|)^2} < \frac{1}{2\lambda}$$

λ and a are system parameters, and x is defined in the reputation algorithm.

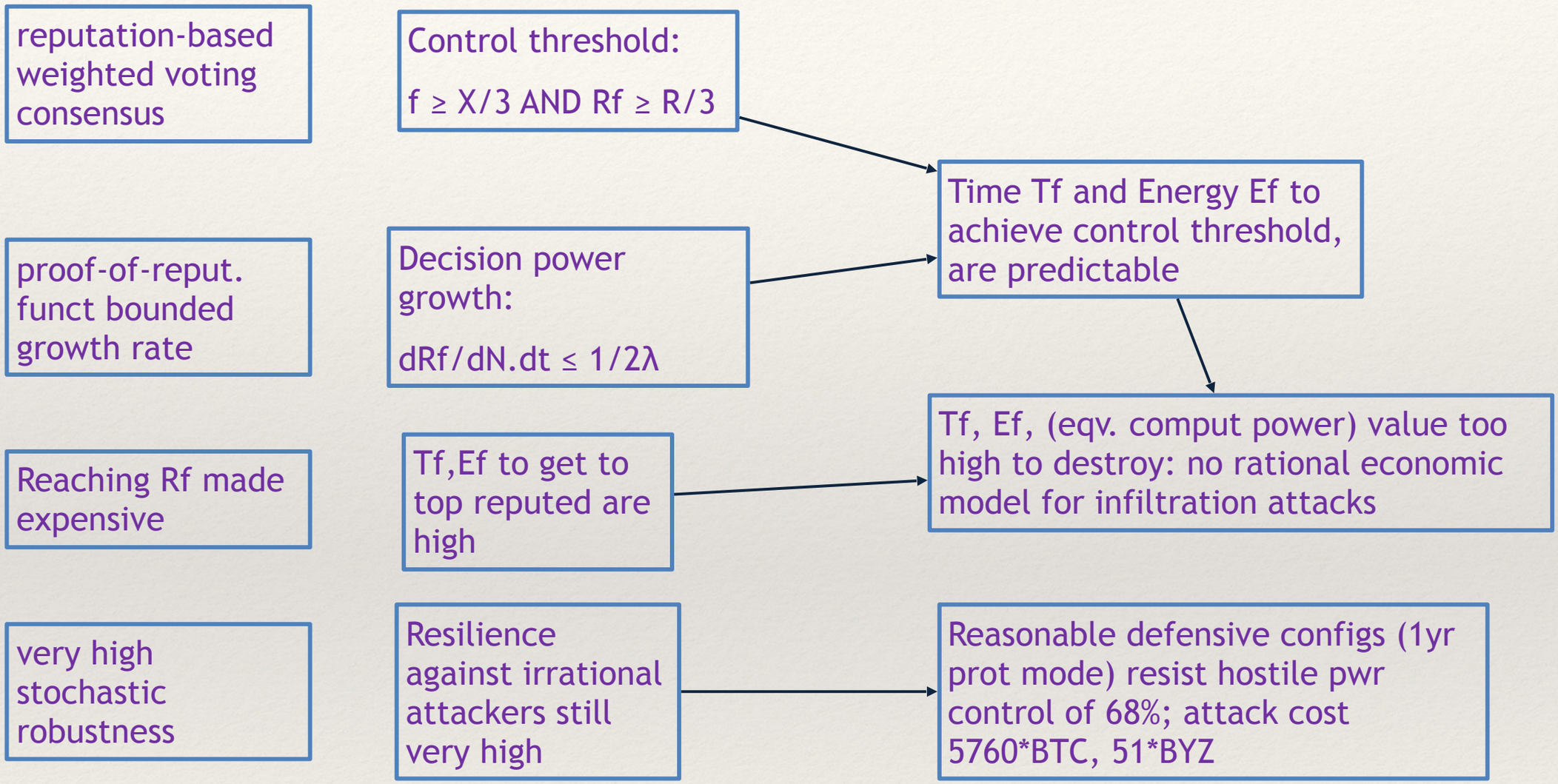
What do we enforce?

The increase of any miner's voting power is bounded by "physics"!

$$\frac{d^2Pd}{dN \cdot dt} = \frac{1}{2}\frac{\lambda}{(\lambda + |x - a|)^2} \leq \frac{1}{2\lambda}$$

λ and a are system parameters, and x is defined in the reputation algorithm.

RECAP: The logic of RepuCoin in a nutshell



MONASH University

Security and Dependability:

The minimum cost of successfully attacking RepuCoin

Joining time\ Target	1 week	1 month	3 months	6 months
1 month	infeasible	45%	30%	27%
3 months	infeasible	90%	45%	33%
6 months	infeasible	infeasible	68%	45%
9 months	infeasible	infeasible	90%	54%
12 months	infeasible	infeasible	infeasible	68%
18 months	infeasible	infeasible	infeasible	91%
20 months	infeasible	infeasible	infeasible	infeasible

Security and Dependability:

The minimum cost of successfully attacking RepuCoin

Joining time\ Target	1 week	1 month	3 months	6 months
1 month	infeasible	BTC: *635; BYZ: *6	BTC: *1271; BYZ: *11	BTC: *2287; BYZ: *20
3 months	infeasible	BTC: *1270; BYZ: *11	BTC: *1906; BYZ: *17	BTC: *2795; BYZ: *25
6 months	infeasible	infeasible	BTC: *2880; BYZ: *26	BTC: *3812; BYZ: *34
9 months	infeasible	infeasible	BTC: *3812; BYZ: *34	BTC: *4574; BYZ: *41
12 months	infeasible	infeasible	infeasible	BTC: *5760; BYZ: *51
18 months	infeasible	infeasible	infeasible	BTC: *7708; BYZ: *69
20 months	infeasible	infeasible	infeasible	infeasible

Comparison

Attacks/Features	BitCoin	BitCoin-NG	ByzCoin	RepuCoin
Double spending attacks	☠	☠	👌	👌
Selfish mining attack	☠	☠	☠	👌
Bribery/flash attack	☠	☠	☠	👌
Eclipse attacks	☠	☠	😐	😐
Non-forkable chain	☠	☠	👌	👌
Liveness	👌	👌	☠	👌
Throughput	7 tps	?	1,000 tps	10,000 tps

256 Bytes / TX
13 nodes
1KB / Kblock
2 MB / Mblock

👌 The system is secure against this attack

☠ The system is vulnerable to this attack

😐 The system can prevent double spending, but its throughput maybe reduced.

Future work: Join me!

MONASH University