Rationality & Altruism
My Favorite Topics
Consensus

\[ n > 3f \]

**PiChain: When a Blockchain meets Paxos**

Conrad Burchert¹ and Roger Wattenhofer²

ETH Zurich, Switzerland
bconrad@ethz.ch
ETH Zurich, Switzerland
wattenhofer@ethz.ch

**Abstract**

We present a new fault-tolerant distributed state machine to inherit the benefits of Paxos, providing strong consistency, and a blockchain-based replication mechanism. In addition, our proposal has a deterministic algorithm to solve Byzantine agreement. We show that consensus on preferences, which we call Preferential Byzantine agreement, introduces a generalization of Byzantine agreement where the values of the nodes are preference rankings instead of boolean values. We show that consensus on preferences, which we call Preferential Byzantine agreement, introduces a generalization of Byzantine agreement where the values of the nodes are preference rankings instead of boolean values.
“Layer 2”

A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels

Christian Decker
Roger Wattenhofer

decher@ethz.ch
wattenhofer@ethz.ch

Algorithmic Channel Design

Georgia Avarikioti
ETH Zurich, Switzerland
zetavar@ethz.ch

Yuyi Wang
ETH Zurich, Switzerland
ywwang@ethz.ch

Roger Wattenhofer
ETH Zurich, Switzerland
wattenhofer@ethz.ch
This Talk

Permissionless Blockchains
Multiple Participants?

- **Yes**
  - Participants Known?
    - **Yes**
      - Permissioned Blockchain
    - **No**
      - Permissionless Blockchain
- **No**
  - No Blockchain (use database)
Do You Trust the Miners?
IL BUONO  IL BRUTTO  IL CATTIVO

CLINT EASTWOOD  ELI WALLACH  LEE VAN CLEEF
Modeling Distributed Systems

Altruistic  Rational  Crash  Byzantine
Modeling Distributed Systems

Who are the Miners?
The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.”
Mining is a Rational Business
Mining is a Rational Business

Japanese Cryptocurrency Monacoin Hit by Selfish Mining Attack
Selfish Mining Timeline

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

2009  2010  2013  2018

Japanese Cryptocurrency Monacoin Hit by Selfish Mining Attack

I came across an idea that I think is worth discussing. I'm calling this a "mining cartel attack". I have no idea what describing it as I'm sure the thought has crossed people's minds before, but I think there are in place to stop this.
What is Selfish Mining
Majority is not Enough: Bitcoin Mining is Vulnerable

Ittay Eyal and Emin Gün Sirer

Department of Computer Science, Cornell University
ittay.eyal@cornell.edu, eg@systems.cs.cornell.edu
### Algorithm 1: Selfish-Mine

1. **on** Init
   - public chain $\leftarrow$ publicly known blocks
   - private chain $\leftarrow$ publicly known blocks
   - $\text{privateBranchLen} \leftarrow 0$
   - Mine at the head of the private chain.

2. **on** My pool found a block
   - $\Delta_{\text{prev}} \leftarrow \text{length(private chain)} - \text{length(public chain)}$
   - append new block to private chain
   - $\text{privateBranchLen} \leftarrow \text{privateBranchLen} + 1$
   - **if** $\Delta_{\text{prev}} = 0$ and $\text{privateBranchLen} = 2$ **then**
     - publish all of the private chain
     - $\text{privateBranchLen} \leftarrow 0$
   - Mine at the new head of the private chain.

3. **on** Others found a block
   - $\Delta_{\text{prev}} \leftarrow \text{length(private chain)} - \text{length(public chain)}$
   - append new block to public chain
   - **if** $\Delta_{\text{prev}} = 0$ **then**
     - private chain $\leftarrow$ public chain
     - $\text{privateBranchLen} \leftarrow 0$
   - **else if** $\Delta_{\text{prev}} = 1$ **then**
     - publish last block of the private chain
   - **else if** $\Delta_{\text{prev}} = 2$ **then**
     - publish all of the private chain
     - $\text{privateBranchLen} \leftarrow 0$
   - **else**
     - publish first unpublished block in private block.
   - Mine at the head of the private chain.
Algorithm 26.2 Selfish Mining

1: Idea: Mine secretly, without immediately publishing newly found blocks
2: Let \( d_p \) be the depth of the public blockchain
3: Let \( d_s \) be the depth of the secretly mined blockchain
4: if a new block \( b_p \) is published, i.e., \( d_p \) has increased by 1 then
5:   if \( d_p > d_s \) then
6:     Start mining on that newly published block \( b_p \)
7:   else if \( d_p = d_s \) then
8:     Publish secretly mined block \( b_s \)
9:     Mine on \( b_s \) and publish newly found block immediately
10:  else if \( d_p = d_s - 1 \) then
11:     Publish both secretly mined blocks
12:   end if
13: end if
\[ d_p > d_s \]
Somewhat Simpler Algorithm

Algorithm 26.2 Selfish Mining

1: Idea: Mine secretly, without immediately publishing newly found blocks
2: Let $d_p$ be the depth of the public blockchain
3: Let $d_s$ be the depth of the secretly mined blockchain
4: if a new block $b_p$ is published, i.e., $d_p$ has increased by 1 then
5:     if $d_p > d_s$ then
6:         Start mining on that newly published block $b_p$
7:     else if $d_p = d_s$ then
8:         Publish secretly mined block $b_s$
9:     else if $d_p = d_s - 1$ then
10:        Publish both secretly mined blocks
11:     end if
12: end if
13: end if
\[ d_p = d_s - 1 \]
Somewhat Simpler Algorithm

**Algorithm 26.2 Selfish Mining**

1. Idea: Mine secretly, without immediately publishing newly found blocks
2. Let $d_p$ be the depth of the public blockchain
3. Let $d_s$ be the depth of the secretly mined blockchain
4. if a new block $b_p$ is published, i.e., $d_p$ has increased by 1 then
5. \hspace{1em} if $d_p > d_s$ then
6. \hspace{2em} Start mining on that newly published block $b_p$
7. \hspace{1em} else if $d_p = d_s$ then
8. \hspace{2em} Publish secretly mined block $b_s$
9. \hspace{2em} Mine on $b_s$ and publish newly found block immediately
10. \hspace{1em} else if $d_p = d_s - 1$ then
11. \hspace{2em} Publish both secretly mined blocks
12. \hspace{1em} end if
13. end if
\[ d_p = d_s \]
α: probability that selfish miner finds a block
Stationary Distribution

\[ p_1 = \alpha p_0 \]

\[ \beta p_{i+1} = \alpha p_i, \text{ for all } i > 1 \]

and \( 1 = \sum_i p_i. \)
Computation...

\[ p_1 = \alpha p_0 \]
\[ \beta p_{i+1} = \alpha p_i, \text{ for all } i > 1 \]
and \[ 1 = \sum_i p_i. \]

Using \( \rho = \alpha / \beta \), we express all terms of above sum with \( p_1 \):

\[ 1 = \frac{p_1}{\alpha} + p_1 \sum_{i \geq 0} \rho^i = \frac{p_1}{\alpha} + \frac{p_1}{1 - \rho}, \text{ hence } p_1 = \frac{2\alpha^2 - \alpha}{\alpha^2 + \alpha - 1} \]
All $\beta$ Transitions

$0 \rightarrow 0$: Block for honest miners

$i + 1 \rightarrow i$: Block for selfish miner (for $i > 2$)

$2 \rightarrow 0$: Two blocks for selfish miner

$1 \rightarrow 0$: Race who wins next block

- with probability $\alpha$ two blocks for selfish miner
- with probability $\beta(1 - \gamma)$ two blocks for honest miners
- with probability $\beta\gamma$ one block each

$\gamma$: probability that honest miners append block to selfish miner’s block (in race)
Ratio of Selfish Blocks in Chain

\[
\frac{1 - p_0 + p_2 + \alpha p_1 - \beta (1 - \gamma) p_1}{1 + p_1 + p_2}
\]

\(\gamma\) : probability that honest miners append block to selfish miner’s block (in race)
Selfish Miner Share

\[
\frac{\alpha(1 - \alpha)^2(4\alpha + \gamma(1 - 2\alpha)) - \alpha^3}{1 - \alpha(1 + (2 - \alpha)\alpha)}
\]
Selfish Miner Share

\[
\alpha(1 - \alpha)^2 \left(4\alpha + \gamma(1 - 2\alpha)\right) - \alpha^3 \quad \frac{1}{1 - \alpha(1 + (2 - \alpha)\alpha)}
\]

\(\gamma = 0\): break even at \(\alpha = 1/3\)

\(\gamma = 0.5\): break even at \(\alpha = 1/4\)

\(\gamma = 1\): break even at \(\alpha > 0\)
A Blockchain Without Altruism?
What about FruitChains?

FruitChains: A Fair Blockchain

Rafael Pass
Cornell Tech
rafael@cs.cornell.edu

Elaine Shi
Cornell University
elaine@cs.cornell.edu
FruitChains in a Single Slide
Are FruitChains Rational?

I work a little extra, so that others get paid.

Why include other fruit? Try double-spending? No punishments!

(Also: fees)
What We Really Want!
A Blockchain Without Altruism?

[Joint Work with Jakub Sliwinski]
Simple Chains Are Too Simple
Better: Expose Competition
Our Rational Blockchain
Always Refer to All Childless Blocks
Only One Type of Reference

(Heaviest Reference is Your “Parent”)
Block Ordering is Recursive

Inclusive Block Chain Protocols

Yoad Lewenberg\textsuperscript{1}, Yonatan Sompolinsky\textsuperscript{1}, and Aviv Zohar\textsuperscript{1,2}
Incentives
Why Miners Should Always Refer to All Childless Blocks?
Because of our Block Rewards!
It’s Somewhat Complicated...
Motivating Block Rewards I

Reward = 0.34

Reward = 0.71

Reward = 0.71

Reward = 1
Motivating Block Rewards II

Reward = 0.92

Reward = 0.69
Our Solution

Definition 3 (Penalty Function). Given are a pair of competing branches $\mathcal{B}_X$ and $\mathcal{B}_Y$ where $|\mathcal{B}_X| \geq |\mathcal{B}_Y|$, and a set $E$ of edges between them, such that every block in $\mathcal{B}_Y$ has an incident edge. Then $f$ is defined as follows:

1. $f$ assigns a maximum penalty to all blocks in the smaller branch:
   \[
   \forall B \in \mathcal{B}_Y : f(B) = 1.
   \]

2. Each block’s penalty is divided among incident edges:
   \[
   \left( \forall (A, B) \in E : f((A, B)) \geq 0 \right) \land \left( \forall B \in \mathcal{B}_X \cup \mathcal{B}_Y : f(B) = \sum_{A \in E(B)} f((A, B)) \right).
   \]

3. Differences in penalties between blocks in the bigger branch are minimised:
   \[
   \forall B \in \mathcal{B}_Y : \left( (A_1, B), (A_2, B) \in E \land f((A_1, B)) > 0 \right) \implies f(A_1) \leq f(A_2).
   \]

Definition 4 (Reward Scheme). Creator of any block $B$ receives an amount $r(B)$ of cryptocurrency to the address $c_B$. Any spending transaction from this address is valid only if included in a block $C$ such that $\text{LCA}(B, C) > 2p$.

\[
    r(B) = R(1 - \max_{B_X, B_Y \in \mathcal{B}_X \cup \mathcal{B}_Y} \sum_{E(B)} f(B)) + \sum_{tx \in T_B} \text{fee}_B(tx)
\]

Here, $R$ is the base block reward, and $E$ consists of edges from the conflict graph of $G$. $\text{fee}_B(tx)$ is discussed in section 3.1.
Block Penalty Example
Block Penalty Example

Diagram: A network consisting of three nodes with the following connections and values:
- Node 1 is connected to nodes 2 and 3.
- Nodes 2 and 3 are connected to each other with a value of 0.5.
- The value on node 1 is 1.
- The values on nodes 2 and 3 are both 0.5.
Block Penalty Example

![Diagram showing a network with nodes labeled 1 and 0.66, 0.5, and connections between them.](image)
The Penalty Algorithm
The Penalty Algorithm
The Penalty Algorithm
Transaction Fees
A → B
fee = 10

A → B
fee = 10
“The system is secure as long as rational honest nodes collectively control more CPU power than any cooperating group of attacker nodes.”
Thank You!
Questions & Comments?

Thanks to Jakub Sliwinski

www.disco.ethz.ch