Lay Down the Common Metrics

Evaluating PoW Consensus Protocols' Security

Ren Zhang
ren@nervos.org
@nirenzang

Bart Preneel
bart.preneel@esat.kuleuven.be
CONFLUX
PUBLISH OR PERISH
TORTOISE AND HARES
BYZCOIN
GOSHAWK
BAHACK'S IDEA
BITCOIN-NG (AETERNITY, WAVES)
BITCOIN'S NAKAMOTO CONSENSUS
ETH madon POW
DECOR+ (ROOTSTOCK)
GHOST-DAG SPECTRE CHAINWEB
FRUITCHAINS PHANTOM BOBTAIL
THE INCLUSIVE PROTOCOL GHOST
CONFLUX
bitcoin’s Nakamoto Consensus

**NC**

- **To resolve fork**
  - Longest chain (roughly) if there is one
  - First-received in a tie

- **To issue rewards**
  - Main chain blocks ▲ receive full rewards
  - Orphaned blocks △ receive nothing

**Key Weakness**

- Imperfect chain quality:
  A <50% attacker can modify the blockchain with high success rate
Imperfect Chain Quality ➤ 3 Attacks

Selfish Mining

The attacker gains **unfair** block rewards; rational miners would join the attacker, which damages decentralization.
The attacker gets the product without paying for it

Double-spending

Tx1: A → Merchant

Merchant delivers the product

Tx2: A → A'

The public

breath time

attacker block

imperfect Chain Quality 3 Attacks

Time
Imperfect Chain Quality ➡️ 3 Attacks

Censorship (feather-forking)

Threat: I will try to invalidate all blocks confirming these txs

“I do not stand by in the presence of evil”

Rational choice: join the attacker in censorship
The attacker becomes a *de facto* owner
A protocol claims to be more secure than NC:

it either
- achieves better chain quality ❶ ❷

or
- resists better against all three attacks:
  - selfish mining ➔ incentive compatibility ❶
  - double-spending ➔ subversion gain ❶
  - censorship ➔ censorship susceptibility ❷

(choose the paper for the math definitions)

Our Evaluation Framework: 4 Metrics

1. profit-driven adversary
2. byzantine adversary
Better-than-NC Candidates

Better-chain-quality protocols

“I can raise the chain quality”
- **UTB**: Ethereum PoW, Bitcoin-NG (Aeternity, Waves)
- **SHTB**: DECOR+ (Rootstock)
- **UDTB**: Byzcoin, Omniledger
- **Publish or Perish**

Attack-resistant protocols

“I don’t need to raise the chain quality, I can defend against the attacks”
- **Reward-all** (“compensate the losers”): Fruitchains, Ethereum PoW, Inclusive, SPECTRE, PHANTOM, ...
- **Punishment** (“fine all suspects”): DECOR+, Bahack’s idea
- **Reward-lucky** (content-based reward): Subchains, Bobtail

In this talk
Check the paper
**MDP-based Method**

**Main idea**
Model the protocol execution as a Markov decision process (MDP), enumerate all the attacker’s reasonable strategies, find the ones that optimize the metrics

**Step 1**
Define the attacker’s utility according to the security metric of interest. e.g., in selfish mining:

\[
\text{utility} = \frac{\text{attacker’s rewards}}{\text{all the rewards}}
\]

**Step 2**
Model the protocol as an MDP
### MDP-based Method

**Step 3**  Solve the MDP, compute the attacker’s **optimal strategies** and their maximum utilities in various settings

**Step 4**  Compare the utilities with NC, find out **when they are better/worse**

**Step 5**  Check the respective strategies, find out **why**
Do not equate the security of a consensus protocol with its cryptocurrency

- Many real-world factors affect the attack difficulty (e.g., 51% attack against ETC vs. against Bitcoin)
- Several systems rely on extra protection for certain attack resistance
The Evaluation Results
**Simplified Results**

<table>
<thead>
<tr>
<th>“Better-chain-quality”</th>
<th>Chain Quality</th>
<th>“Attack-resistant”</th>
<th>Incentive compatibility</th>
<th>Subversion gain</th>
<th>Censorship susceptibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform tie-breaking</td>
<td>😞</td>
<td>Reward-all</td>
<td>😞</td>
<td>😞</td>
<td>🤚</td>
</tr>
<tr>
<td>Smallest-hash tie-breaking</td>
<td>😞</td>
<td>Punishment</td>
<td>😞</td>
<td>😞</td>
<td>🤚</td>
</tr>
<tr>
<td>Unpredictable deterministic tie-breaking</td>
<td>😞</td>
<td>Reward-all</td>
<td>😞</td>
<td>😞</td>
<td>🤚</td>
</tr>
<tr>
<td>Publish or perish</td>
<td>😞</td>
<td>Reward-lucky</td>
<td>😞</td>
<td>😞</td>
<td>🤚</td>
</tr>
</tbody>
</table>

Better-than-chain-quality: it depends
Worse-than-chain-quality: 💩
*Attack-Resistant Reward-All: Fruitchains*

- **Same mining procedure, two products:**
  - A block if the first $k$ bits of $H(\text{candidate}) < D_1$
  - A fruit if the last $k$ bits of $H(\text{candidate}) < D_2$

- Fruits in blocks; txs in fruits
- Fork-resolving: longest chain + first received (same as NC, RS and Subchains)

---

**Diagram:**

- **A**
- **B**
- **C**
- **D**
- **E**

- Parent block
- Time

**Flow:**

- A -> B
- B -> C
- C -> D
- E

**Symbols:**

- A: Apple
- B: Banana
- C: Blueberry
- D: Date
- E: Mixed fruits (Apple, Banana, Blueberry)
Each fruit has a **pointer block**: a recent block the fruit miner is sure will not be orphaned.

A fruit is validity if:  
- The pointer block is in the main chain (sorry tomato).  
- Gap(fruit)=height(host)-height(pointer) < TimeOut
  (If TimeOut=3, pear is hopeless).

**Reward distribution**  
- Valid fruits receive rewards; blocks, nothing.
Fruitchains Results

📍 Incentive compatibility & Subversion Gain

 риск

Risk-free units -> more audacious behaviors: attacker uses worthless blocks to invalidate honest fruits; attacker’s first fruits are in both chains

Incentive compatibility & Subversion Gain

Risk-free units -> more audacious behaviors: attacker uses worthless blocks to invalidate honest fruits; attacker’s first fruits are in both chains
Fruitchains Results

- Censorship
- Susceptibility

- Fruits in invalidated blocks might be added back later (lucky orange)
An uncle is valid if
- \( \text{Gap(uncle)} = \text{height(host)} - \text{height(uncle)} < \text{TimeOut} \)
  
- \( B' \) is hopeless if \( \text{TimeOut} = 3 \)

- Each block reward is \textit{evenly split} among competing block & uncles of the same height

(RS is modified from DECOR+, but their results are not the same!)
3-confirmation RS performs better than 9-conf. Fruitchains

Min double-spending reward to incentivize double-spending attack attempts

Attacker controls 10% mining power, 6-conf., bounty = 102 block rewards in NC, 346 in RS, 0 in Fruitchains
Censorship Susceptibility of RS

weak attackers

strong attackers

\[ \text{Gap} = h(\text{host}) - h(\text{self}) \]
When chain quality is not perfect ...

- Reward all -> no risk to double-spend
- Punish -> aid censorship
- Reward lucky -> lucky ≠ good

Need to go beyond reward distribution policy to solve all attacks
Simplicity is beauty

What not to do

- No protocol comprehensively outperforms NC
- Designing protocols too complicated to analyze
- Security analysis
  - against one attack strategy
  - against one attacker incentive
  - with unrealistic parameters
Discussion

Better chain quality & attack resistance?

Practical assumptions
- Awareness of network conditions
- Loosely synchronized clock
- Real-world commitments

Outsource liability to raise attack resistance
- Introduce additional punishment rules (embed proofs of malicious behavior in blockchain)
- Solve at layer 2 (e.g. lightning guarantees double spending resistance)
Tell anyone that claims to have a perfectly secure consensus protocol...
ACADEMIA IS WATCHING YOU
Thank you!

Code: github.com/nirenzang/PoWSecurity

Ren Zhang  Bart Preneel
ren@nervos.org  bart.preneel@esat.kuleuven.be
@nirenzang