Cryptocurrencies and (PoW) Distributed Consensus

Ren Zhang & Bart Preneel
ren.zhang@esat.kuleuven.be
bart.preneel@esat.kuleuven.be

1

Science of Nakamoto Consensus

[Garay-Kiayias-Leonardos'15] [Kiayias-Panagiotakos'15] [Pass-Seeman-Shelat17]

- **chain growth:** chain grows proportionally with the number of time steps
- **chain quality/blockchain quality/fairness:** fraction of blocks proportional to mining power
- **(blockchain) consistency:** agreement among players on blockchain except for last σ blocks
- **liveliness:** no transaction censorship

2

Science of Nakamoto Consensus

[PSS17] Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. Eurocrypt'17

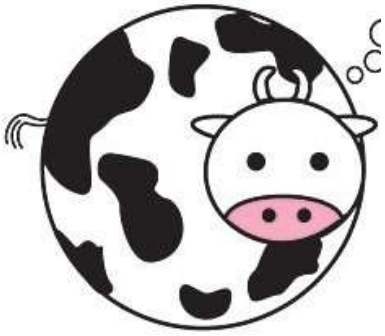
Theorem 2.6 (Security of Nakamoto [PSS17]). *For any constant $\delta > 0$, any $0 < p < 1$, any super-logarithmic function $T_0 = \omega(\log \kappa)$ Nakamoto's blockchain protocol $\Pi_{\text{nak}}(p)$ satisfies the following properties in Γ^p_{nak}-environments:*

- *T_0-consistency;*
- *chain growth rate (T_0, g_0, g_1) where*

$$g_0 = (1-\delta)\gamma$$
$$g_1 = (1+\delta)np$$

- *chain quality (T_0, μ) where*

$$\mu = 1 - (1+\delta)\frac{\beta}{\gamma}$$



3

Tortoise and Hares
Publish or Perish
SUBCHAINS
BYZCOIN
GOSHAWK
BAHACK'S IDEA
BITCOIN-NG (AETERNITY, WAVES)
BITCOIN'S NAKAMOTO CONSENSUS
ETHEREUM PoW
DECOR+ (ROOTSTOCK)
GHOST-DAG
SPECTRE
CHAINWEB
FRUITCHAINS
PHANTOM
GHOST
BOBTAIL
THE INCLUSIVE PROTOCOL
Conflux

4

5

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

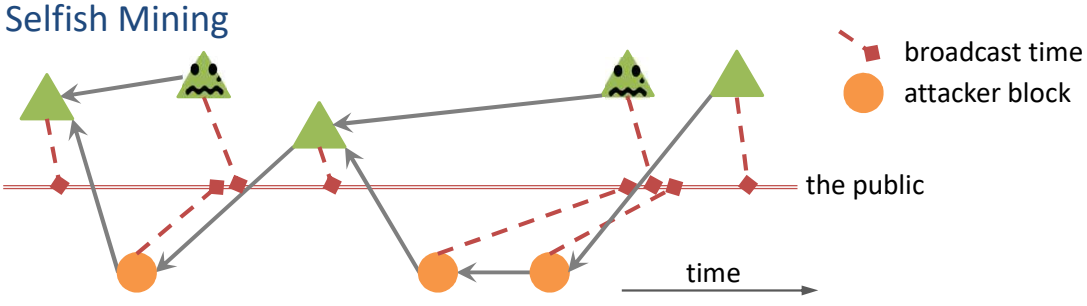
- imperfect chain quality:
  a <50% attacker can modify the blockchain with high success rate

6

7

Imperfect Chain Quality ☞ 3 Attacks

Selfish Mining

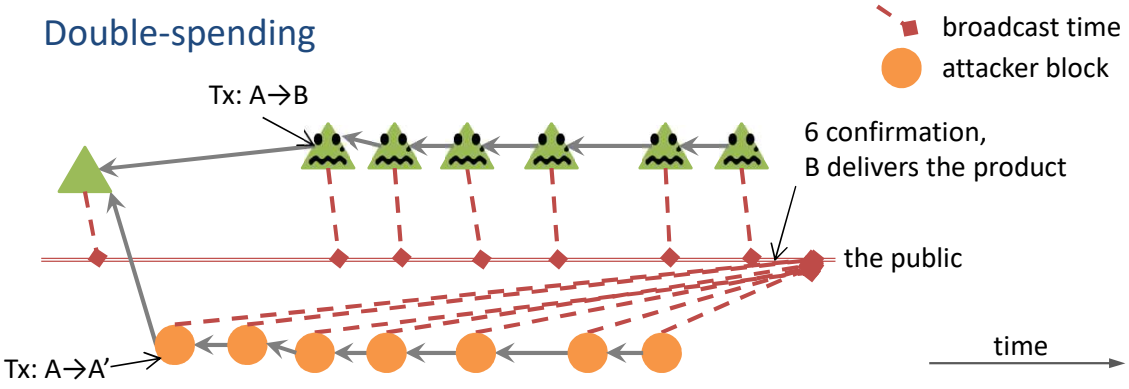


The attacker gains unfair block rewards; rational miners would join the attacker, which damages decentralization

8

Imperfect Chain Quality ☞ 3 Attacks

Double-spending



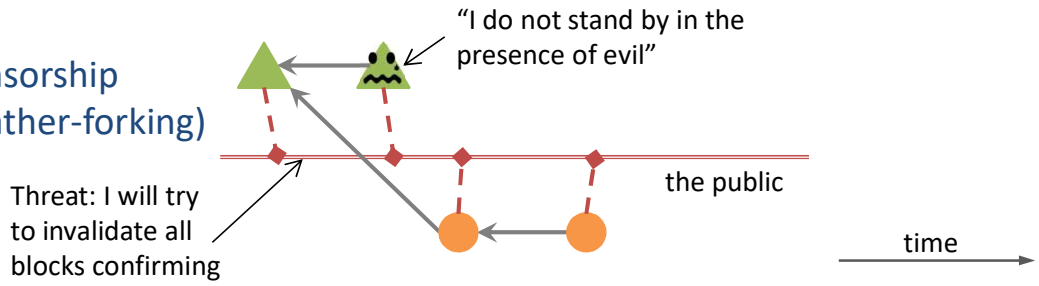
The attacker reverses confirmed txs

Subversion bounty = minimum double-spending reward to incentivize attack attempts

9

Imperfect Chain Quality ☞ 3 Attacks

Censorship (feather-forking)



Rational choice: join the attacker in censorship
The attacker becomes a *de facto* owner

These 3 attacks are most influential.

10

Other attacks
– out of scope as beyond pure consensus protocol

[Bonneau'16]
- Renting mining equipment
- Bribing miners

[Meshkov+'17]
- Coin hopping (based on difficulty adjustments)

[Eyal'15]
[Kwon+'17]
- Attacks on mining pools

[Carlsten+'16]
[Tsabary+'18]
- If block rewards shrink: claim less transaction fees on fork so miners join for remaining higher fees

11

Our Evaluation Framework: Four Metrics

A protocol claims to be more secure than NC:

it either
- achieves better chain quality

or
- resists better against all three attacks:
  - selfish mining ☞ incentive compatibility (revenue)
  - double-spending ☞ subversion gain
  - censorship ☞ censorship susceptibility

Byzantine adversaries rather than rational

(check [Zhang-P'19] for the math definitions)

12

Candidates

Better-chain-quality protocols

[tie breaking rule]

- "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

[topology/reward distribution]

this talk

check [Zhang-P'19]

- "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

13

Attack model

- Attacker works on a single chain
- Ignore transaction fees
- Expected block interval identical for all protocols
- Zero natural orphan rate (low delay)

Longest chain rule + rational attacker: can prove that there are at most two chains: public/attacker

14

MDP-based Method

[Saphirstein-Sompolinsky-Zohar, FC'16]

1. Define the attacker's utility according to the security metric of interest
2. Model the consensus protocol as a Markov decision process (MDP)
3. Compute the attacker's optimal strategies and their maximum utilities in various settings

15

MDP description

S: State space

A: Action space

P: Stochastic transition matrix

R: Reward matrix

16

MDP: Action space A for Bitcoin

a length of attacker's chain after last fork

h blocks of honest miner's chain after last fork

Adopt: attack accepts honest network chain; discard a attacker blocks

Override: attacker publishes his blocks to form longest chain (a > h)

Match: most recent block was published by honest miners; attacker publishes a block to create a tie

Wait: attacker keeps mining

17

MDP: State space for Bitcoin

(a, h, fork)

a length of attacker's chain after last fork

h blocks of honest miner's chain after last fork

fork:

relevant: previous state was of form (a, h-1, *)
(a ≥ h, match is feasible)

irrelevant: previous state was of form (a-1, h, *)
match not feasible

active: honest network is already split due to a match

18

MDP: Transition and reward matrices

Prob. α Initial state is (1,0,irrelevant)

Prob. 1-α Initial state is (0,1,irrelevant)

Reward: (accepted attacker blocks, accepted honest blocks)

State $\times$ Action	State	Probability	Reward
$(a, h, \cdot), adopt$	$(1, 0, irrelevant)$	α	$(0, h)$
	$(0, 1, irrelevant)$	$1 - \alpha$	
$(a, h, \cdot), override^{\dagger}$	$(a - h, 0, irrelevant)$	α	$(h + 1, 0)$
	$(a - h - 1, 1, relevant)$	$1 - \alpha$	
$(a, h, irrelevant), wait$	$(a + 1, h, irrelevant)$	α	(0,0)
$(a, h, relevant), wait$	$(a, h + 1, relevant)$	$1 - \alpha$	(0,0)
$(a, h, active), wait$	$(a + 1, h, active)$	α	(0,0)
$(a, h, relevant), match^{\ddagger}$	$(a - h, 1, relevant)$	$\gamma \cdot (1 - \alpha)$	$(h, 0)$
	$(a, h + 1, relevant)$	$(1 - \gamma) \cdot (1 - \alpha)$	(0,0)

†feasible only when $a \geq h$
‡feasible only when $a \geq h$

19

MDP challenges

Objective function is non-linear

Can only solve for finite state space (size 10^7):

- simplified attack strategies: bounds
- estimate truncation error

20

MDP-based Method

1. Define the attacker's utility according to the security metric of interest
2. Model the consensus protocol as a Markov decision process (MDP)
3. Compute the attacker's optimal strategies and their maximum utilities in various settings
4. Compare the utilities with NC, find out when they are better/worse
5. Check the respective strategies, find out why

21

Cows Are Not Round in Reality

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 introduce extra protection after we started this work

23

Simplified "Better-Chain-Quality" Results

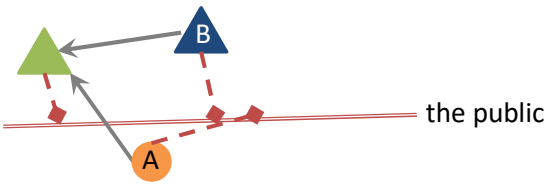
better
it depends
worse

"Better-chain-quality" Protocol	Chain Quality
Uniform tie-breaking Ethereum PoW, Bitcoin-NG (Aeternity, Waves)	(omitted here, check the paper)
Smallest-hash tie-breaking DECOR+ (Rootstock)	?
Unpredictable deterministic tie-breaking DÉCOR+LAMI, Byzcoin, Omniledger	?
Publish or perish	(omitted here, check the paper)

24

Better-Chain-Quality: SHTB & UDTB

γ = fraction of nodes to which attacker can send blocks first (in case of a tie)



Smallest hash tie-breaking (SHTB)

- Compare H(A) and H(B): break the tie with the smallest hash regardless of which one is received first

Unpredictable deterministic tie-breaking (UDTB)

- Compare, e.g., $F_K(A \oplus B, A)$ and $F_K(A \oplus B, B)$: break the tie with a deterministic PRF regardless of which one is received first

NC, γ=0.5

- First received tie-breaking; when two chains broadcast simultaneously, choose randomly

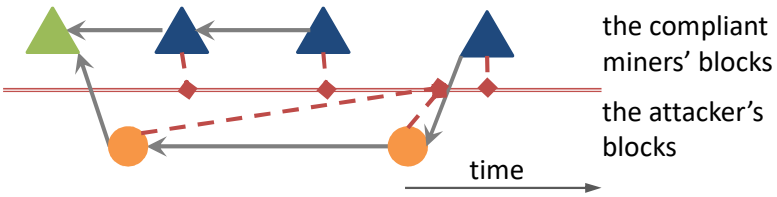
25

Chain Quality of Better-Chain-Quality

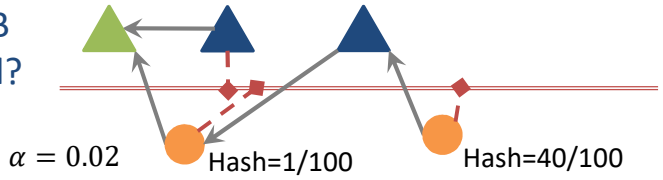
Ranking

NC,$\gamma = 0.5$ > UDTB > SHTB

Why is NC,$\gamma = 0.5$ better than UDTB?

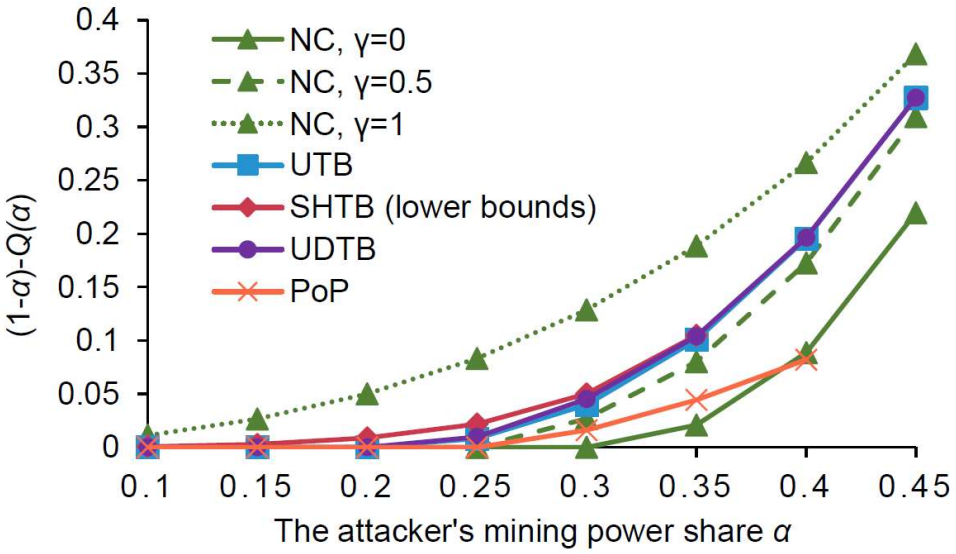


Why does SHTB perform so bad?



26

Simplified "Better-Chain-Quality" Results



Simplified "Better-Chain-Quality" Results

better
it depends
worse

"Better-chain-quality" Protocol	Chain Quality
Uniform tie-breaking Ethereum PoW, Bitcoin-NG (Aeternity, Waves)	😠
Smallest-hash tie-breaking DECOR+ (Rootstock)	😠
Unpredictable deterministic tie-breaking DECOR+LAMI, Byzcoin, Omniledger	😠
Publish or perish	😟

28

Better-Chain-Quality Protocols: General Results

- No protocol achieves the ideal chain quality when the attacker mining power $\alpha > 1/4$
- No protocol performs better than NC, $\gamma = 0$ for all α

Why?

- The protocols cannot distinguish between honest/attacker blocks

Why can't they?

- Information asymmetry: the attacker acts on all info; compliant miners do not

Why don't they?

- Inconsistent assumptions: (try to be) asynchronous, acting on limited public info

29

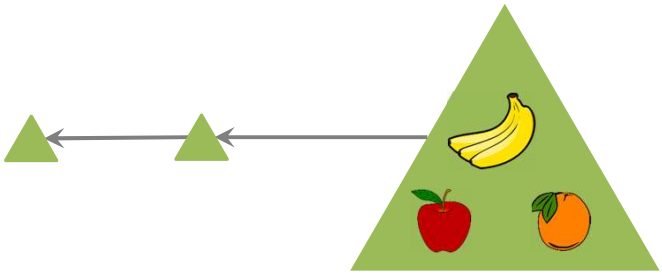
"Attack-Resistant" Results

☺ better
😐 it depends
☹ worse

"Attack-resistant" Protocol	Incentive compatibility	Subversion gain	Censorship susceptibility
Reward-all ☞FruitChains	?	?	?
Punishment ☞Reward-splitting	?	?	?
Reward-lucky ☞Subchains	?	?	?

30

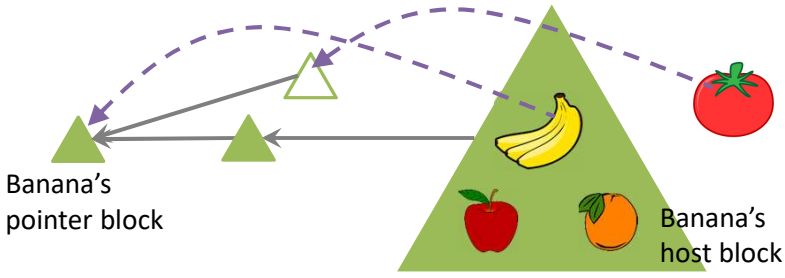
Attack-Resistant☞Reward-All: FruitChains



- Same mining procedure, two products:
  - A block if the first k bits of H(candidate) <D1
  - A fruit if the last k bits of H(candidate) <D2
- Fruits in blocks; txs in fruits
- Fork-resolving: longest chain + first received (same as NC, RS and Subchains)

31

Attack-Resistant☞Reward-All: FruitChains



Why: stop attackers who generate and hide fruits during a long time and publish them at once

- Each fruit has a pointer block: a recent block the fruit miner is sure will not be orphaned
- A fruit is valid if both conditions are met:
  - the pointer block is in the main chain (sorry tomato)
  - Gap(fruit)=height(host)-height(pointer) < TimeOut
- Valid fruits receive rewards; blocks, nothing

32

FruitChains [Pass-Shi'17]

Why selfish mining fails

"[…] even if an adversary tries to erase some block mined by an honest player (which contains some honest fruits), by the chain growth and chain quality properties of the underlying blockchain, eventually an honest player will mine a new block which is stable and this honest player will include the fruits" (and fruit will still be "fresh")

33

FruitChains Results [Pass-Shi'17]

Theorem 4.1 (Security of FruitChain). *For any constant $0 < \delta < 1$, and any p, p_f, let $R = 17$, $\kappa_f = 2qR\kappa$, and $T_0 = 5\frac{\kappa_f}{\delta}$. Then the* FruitChain *protocol denoted $\Pi_{\text{fruit}}(p, p_f, R)$ satisfies*

- *κ_f-consistency;*
- *chain growth rate (T_0, g_0, g_1) where*

$$g_0 = (1-\delta)(1-\rho)np_f,$$

$$g_1 = (1+\delta)np_f$$

- *fairness (T_0, δ).*

in $\Gamma_{\text{fruit}}^{p,p_f,R}$-environments.

No parameters specified

Confirmation time increases with T_0

34

FruitChains Results

better
it depends
worse

"Attack-resistant" Protocol	① Incentive compatibility	② Subversion gain	③ Censorship susceptibility
Fruitchains	😐	😠	😁

①

- Risk-free units -> more audacious behaviors: attacker uses worthless blocks to invalidate honest fruits

②: less financial risk to attack

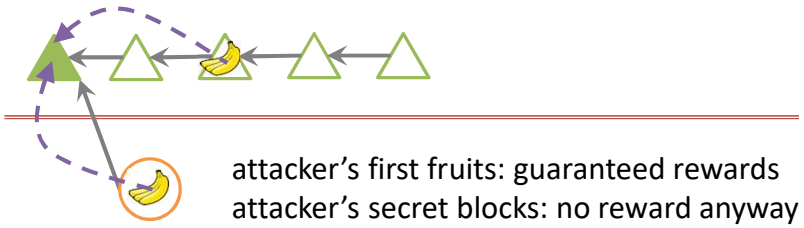
- In NC, a failed double-spending attempt results in losing all block rewards; in FruitChains, the attacker gets the first several fruit rewards

35

FruitChains Results

better
it depends
worse

"Attack-resistant" Protocol	① Incentive compatibility	② Subversion gain	③ Censorship susceptibility
Fruitchains	😐	😠	😁



attacker's first fruits: guaranteed rewards
attacker's secret blocks: no reward anyway

No risk for double-spending!

36

FruitChains Results

😀 better
😐 it depends
😠 worse

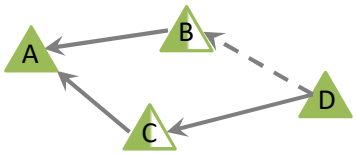
"Attack-resistant" Protocol	① Incentive compatibility	② Subversion gain	③ Censorship susceptibility
Fruitchains	😐	😠	😀

① and ②
Better in ③

- More fruits makes things slightly better
- Fruits in invalidated blocks might be added back later (communication overhead) – unless attacker wins a (long) block race

37

Attack-Resistant☞Punishment: RS



B's host block: D
Gap(B)=1

No pointer, unlike Fruitchains

- An uncle is valid if
  - Gap(uncle)=height(host)-height(uncle) < TimeOut
- Each block reward is evenly split among competing block & uncles of the same height

R(B)=R(C)=0.5R(A)=0.5R(D)

(Note: RS is modified from DECOR+, but their results are not the same!)

38

RS Results

better
it depends
worse

"Attack-resistant" Protocol	① Incentive compatibility	② Subversion gain	③ Censorship susceptibility
Reward-splitting	😀	😀	☹

Better than NC in ① and ②

- 3-confirmation RS performs better than 9-conf. FruitChains (risk of withholding a block)

Subversion Bounty

- If $\alpha = 0.1$, 6 block confirm., subversion bounty = 0 block rewards in Fruitchains, 102 in NC, 346 in RS

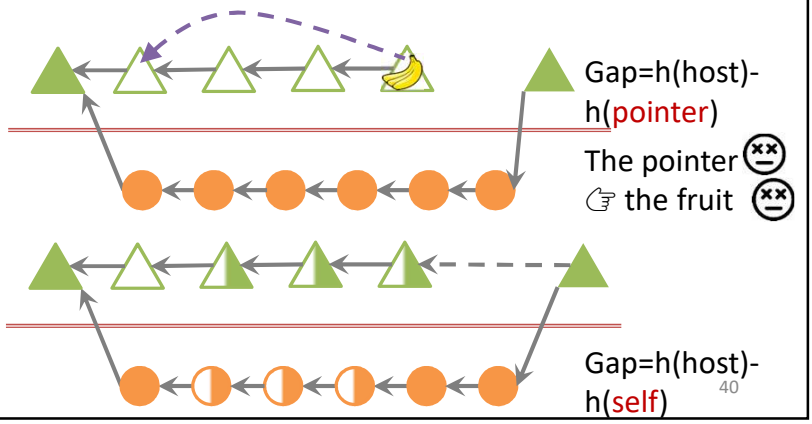
39

Censorship Susceptibility of RS

Rankings

For small α: Fruitchains < NC < RS (not good)
For big α: RS (best), Fruitchains < NC

Why can RS defend against strong attackers?



40

Attack-Resistant☞Reward-Lucky: Subchains

- Same mining procedure, two products:
  - A block if H(candidate)<D1
  - A weak block if D1<H(candidate)<D2
- Weak blocks count in chain length, confirm txs
- Only blocks receive block rewards

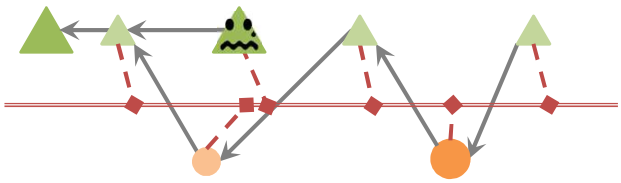
41

Subchains Results

😀 better
😕 it depends
☹ worse

"Attack-resistant" Protocol	① Incentive compatibility	② Subversion gain	③ Censorship susceptibility
Subchains	☹	☹	☹

Worst in ①, ② and ③



weak block

42

Subchains Results

better
it depends
worse

"Attack-resistant" Protocol	① Incentive compatibility	② Subversion gain	③ Censorship susceptibility
Subchains	☹	☹	☹

Worst in ①, ② and ③

- Risk-free units -> more audacious behaviors: Subchains allow attacker to use worthless weak blocks to invalidate honest blocks
- More weak blocks makes things worse

43

Simplified Results

better
it depends
worse

"Better-chain-quality"	Chain Quality
Uniform tie-breaking	☹
Smallest-hash tie-breaking	☹
Unpredictable deterministic tie-breaking	☹
Publish or perish	😐

"Attack-resistant"	Incentive compa-tibility	Subversion gain	Censorship susceptibility
Reward-all ☞Fruitchains	😐	☹	😀
Punishment ☞Reward-splitting	😀	😀	😐
Reward-lucky ☞Subchains	☹	☹	☹

44

Attack-Resistant Protocols: General Results

Security vs. Performance

- Longer confirmation helps
- More bandwidth consumption may help

Dilemma

"Rewarding the bad vs. punishing the good"

- 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

45

Discussion

NC rocks!

- Simplicity is beauty

What not to do

- Designing protocols too complicated to analyze
- Security analysis against one attack strategy
- Security analysis against one attacker incentive
- Security analysis with unrealistic or unspecified parameters

46

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)

47

Want to know more?

- J.A. Garay, A. Kiayias, N. Leonardos, The Bitcoin backbone protocol: Analysis and applications, Eurocrypt'15
- R. Pass, L. Seeman, A. Shelat. Analysis of the blockchain protocol in asynchronous networks. Eurocrypt'17
- A. Sapirshtein, Y. Sompolinsky, and A. Zohar, Optimal selfish mining strategies in Bitcoin, Financial Cryptography and Data Security, 2016.
- R. Zhang, B. Preneel, On the Necessity of a Prescribed Block Validity Consensus: Analyzing Bitcoin Unlimited Mining Protocol, ACM CoNEXT '17
- R. Zhang, B. Preneel, Lay Down the Common Metrics: Evaluating Proof-of-Work Consensus Protocols' Security, IEEE Symposium on Security and Privacy (SP 2019)

48

NERVOS
COSIC

Thank you!

Ren Zhang & Bart Preneel
ren.zhang@esat.kuleuven.be
bart.preneel@esat.kuleuven.be

49