A Rational Protocol Treatment of 51% Attacks

Christian Badertscher[1] **Yun Lu**[2] Vassilis Zikas[3]

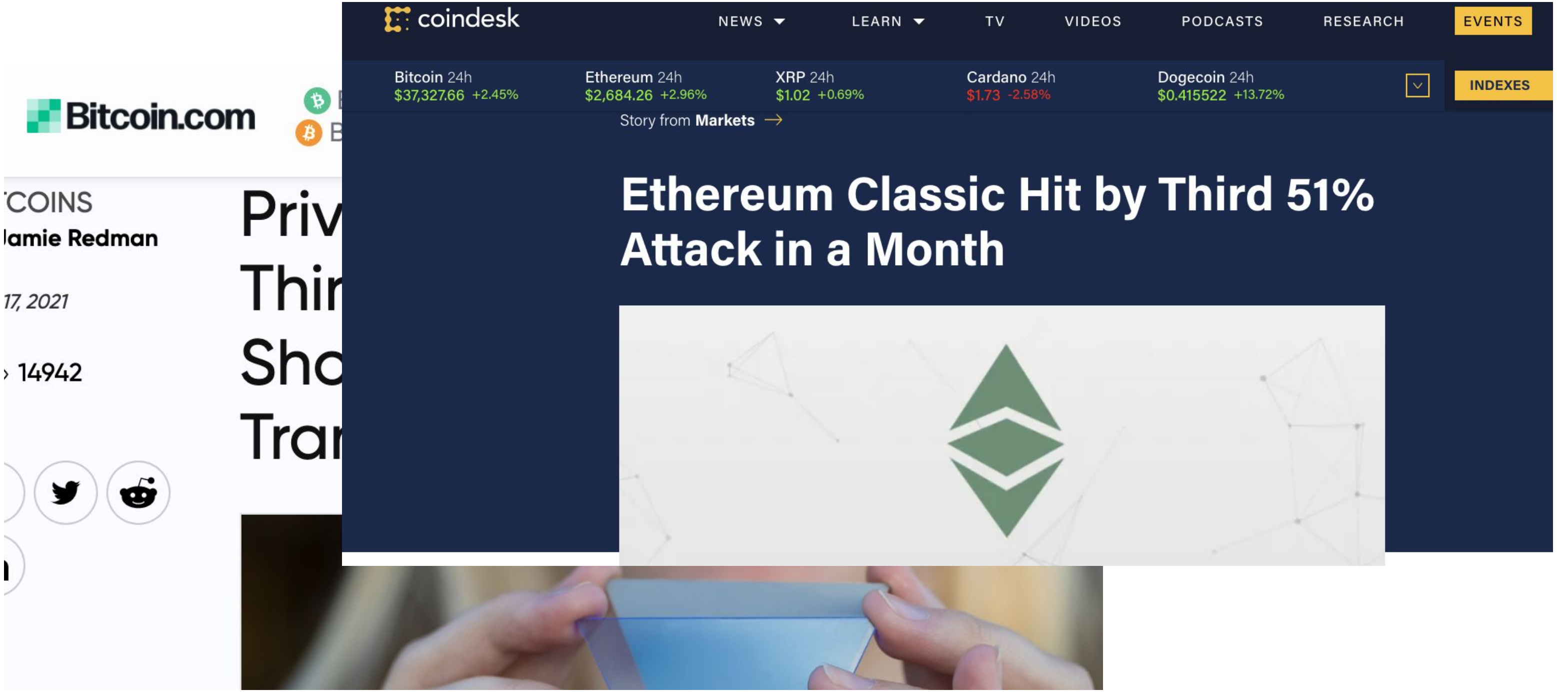
[1] IOHK, [2] University of Edinburgh, [3] Purdue University

https://eprint.iacr.org/2021/897

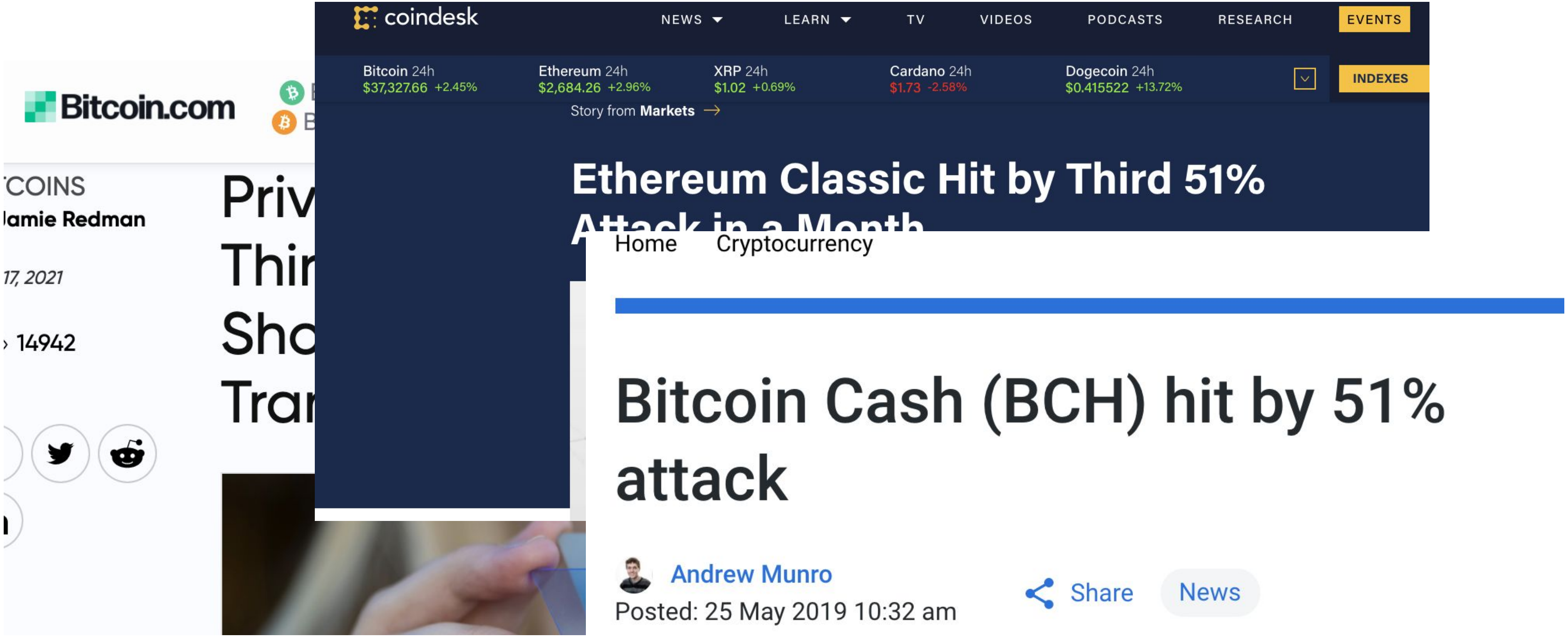
CRYPTO 2021

Crypto on the news

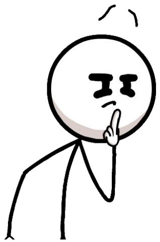
Crypto on the news



Crypto on the news



Exchanging Cryptocurrency for Fiat Currency

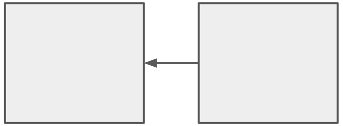


100

Exchange

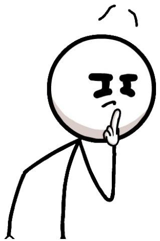
$ 5000

SadCoin Blockchain



Exchanging Cryptocurrency for Fiat Currency

Exchange 100 SadCoins for $5000?

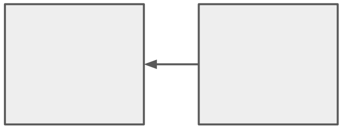


100

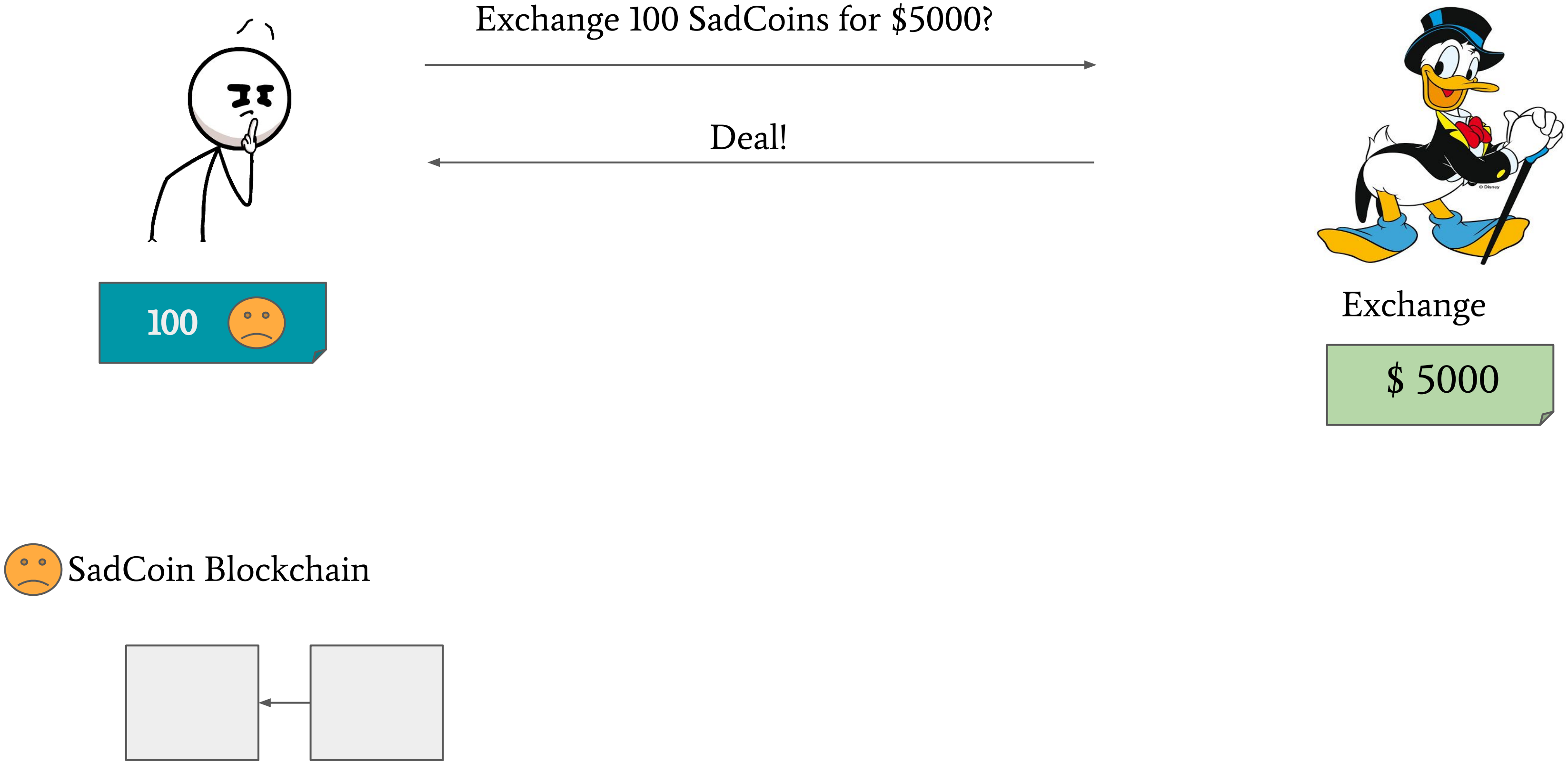
Exchange

$ 5000

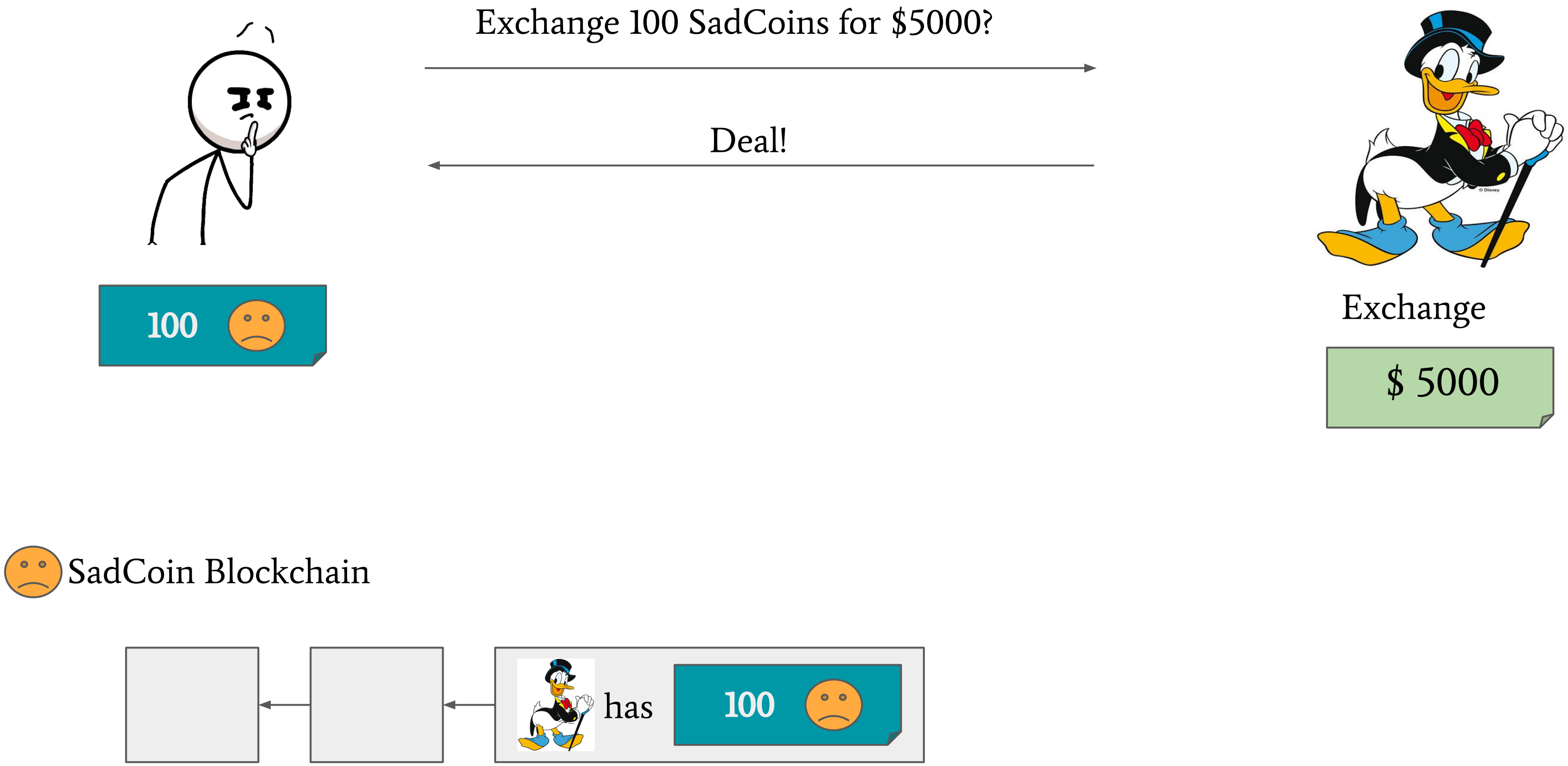
SadCoin Blockchain



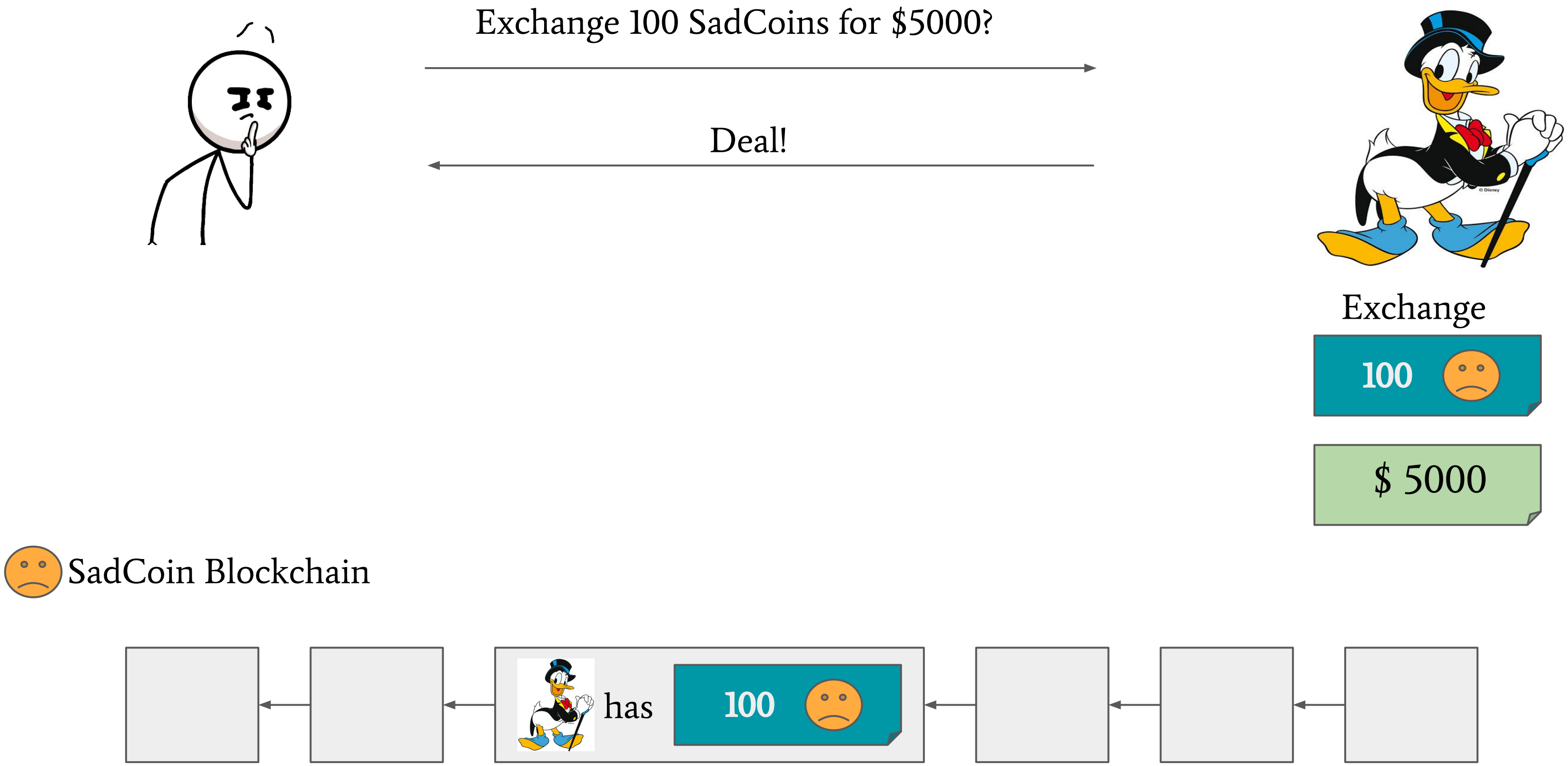
Exchanging Cryptocurrency for Fiat Currency



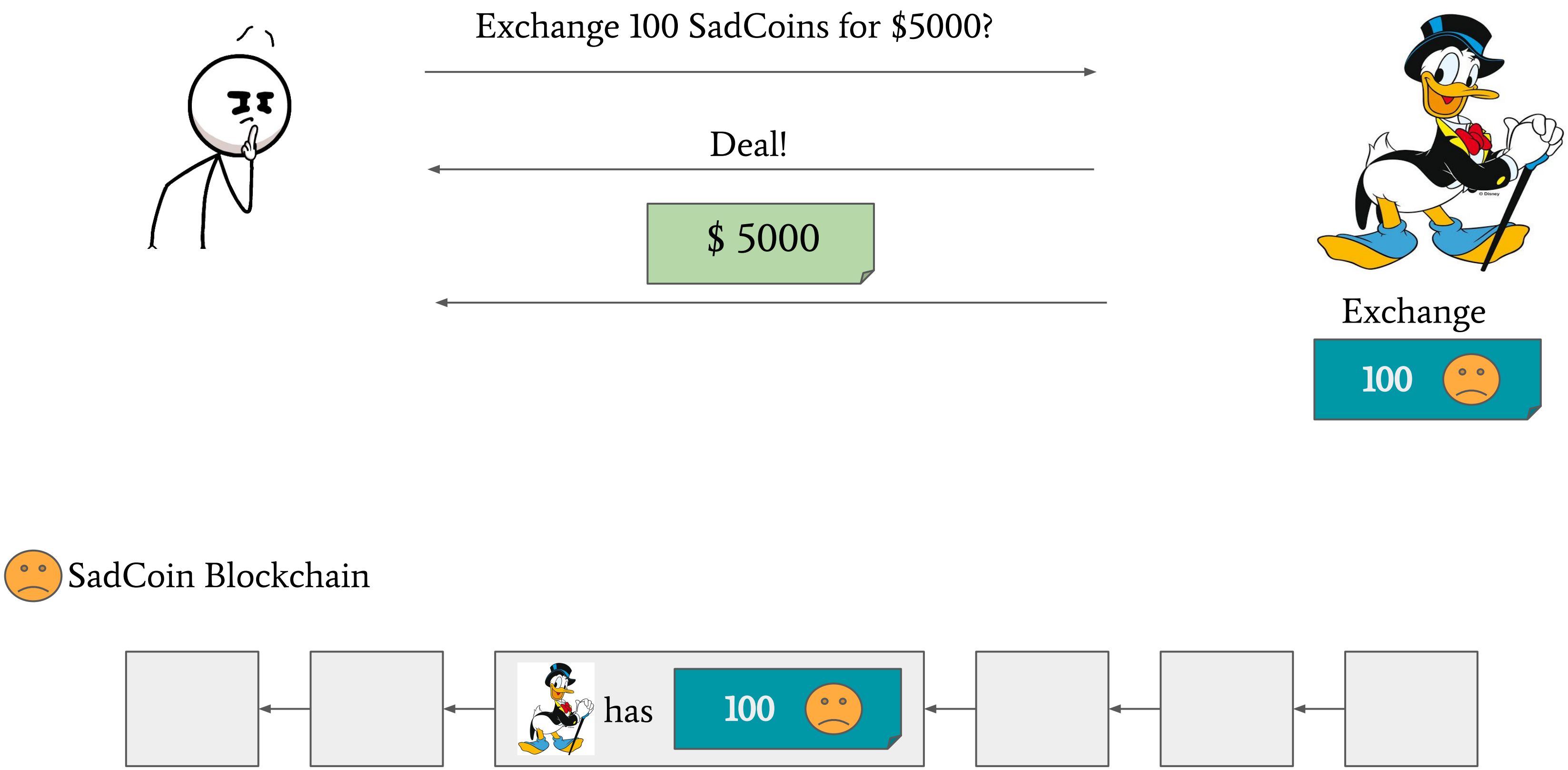
Exchanging Cryptocurrency for Fiat Currency



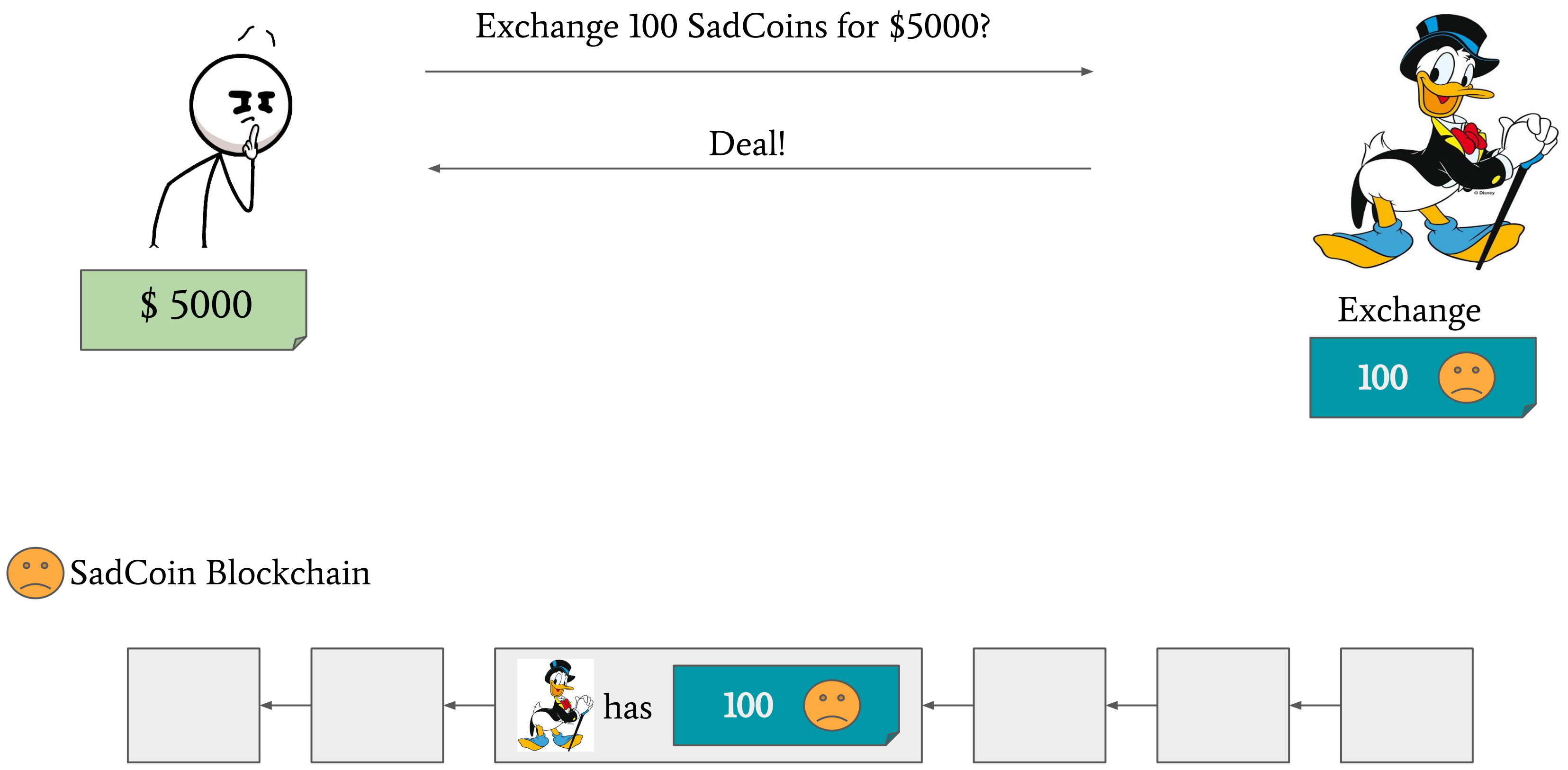
Exchanging Cryptocurrency for Fiat Currency



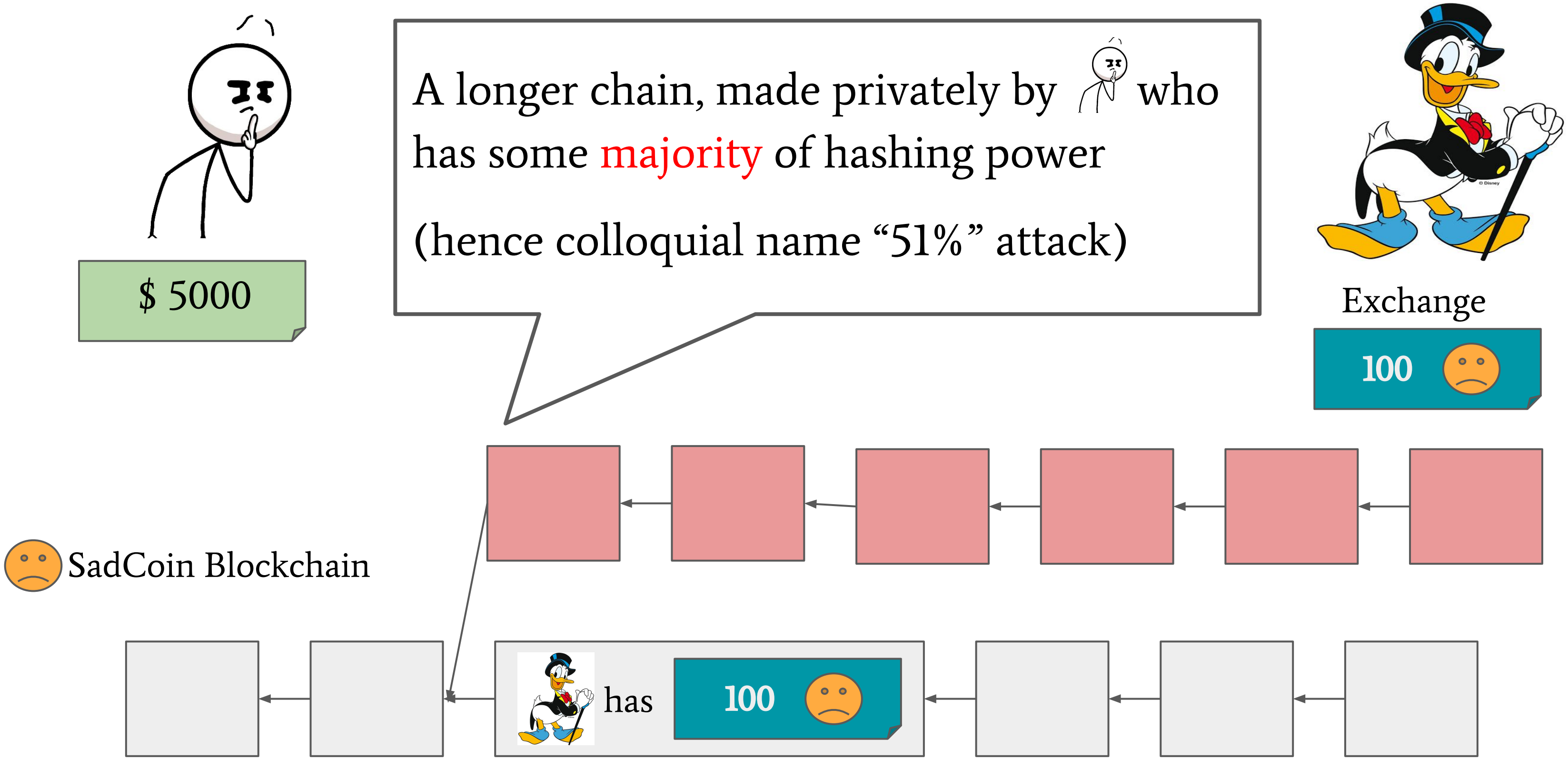
Exchanging Cryptocurrency for Fiat Currency



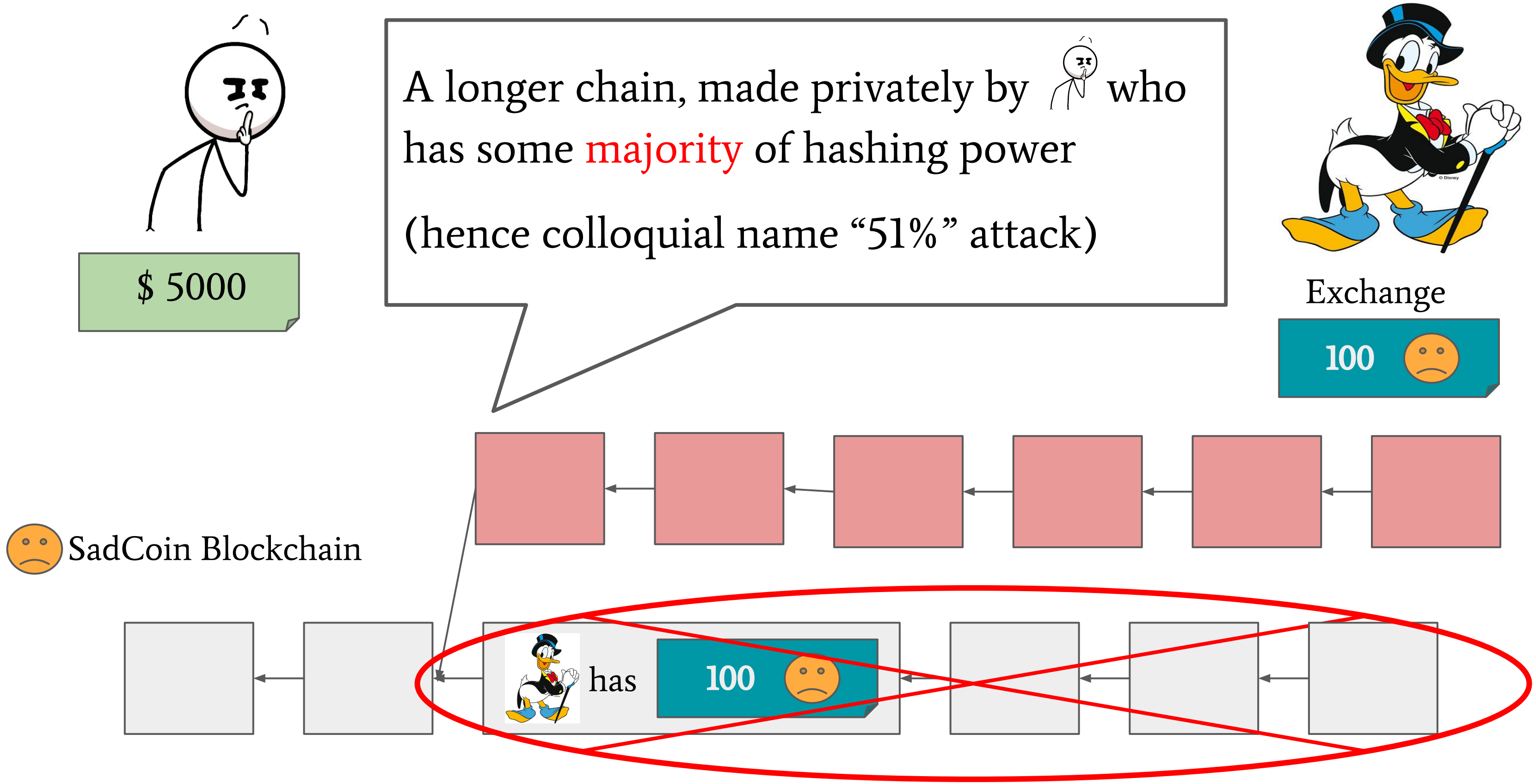
Exchanging Cryptocurrency for Fiat Currency



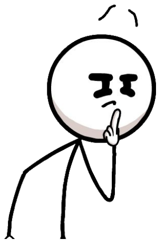
Double-spending



Double-spending



Double-spending



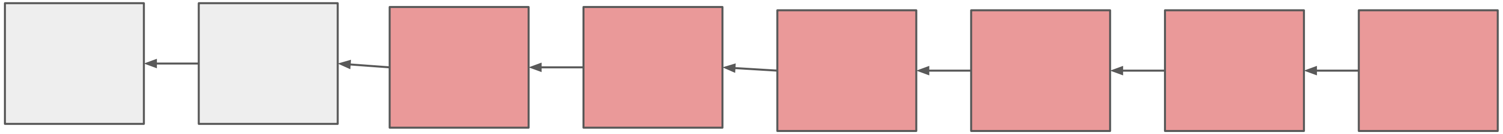
$ 5000

100

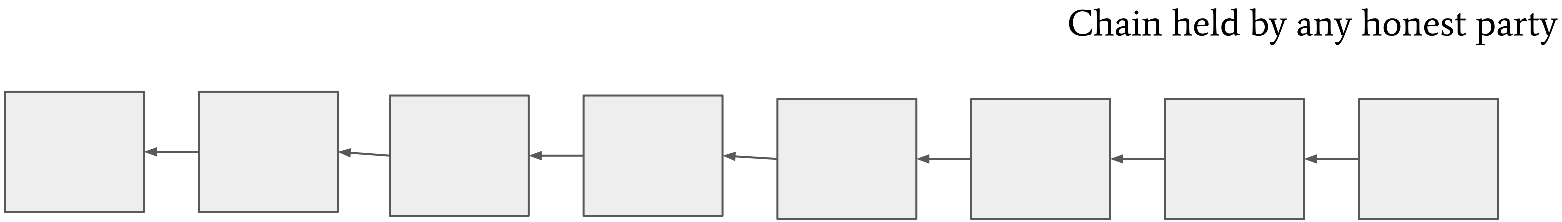
Exchange

100

SadCoin Blockchain



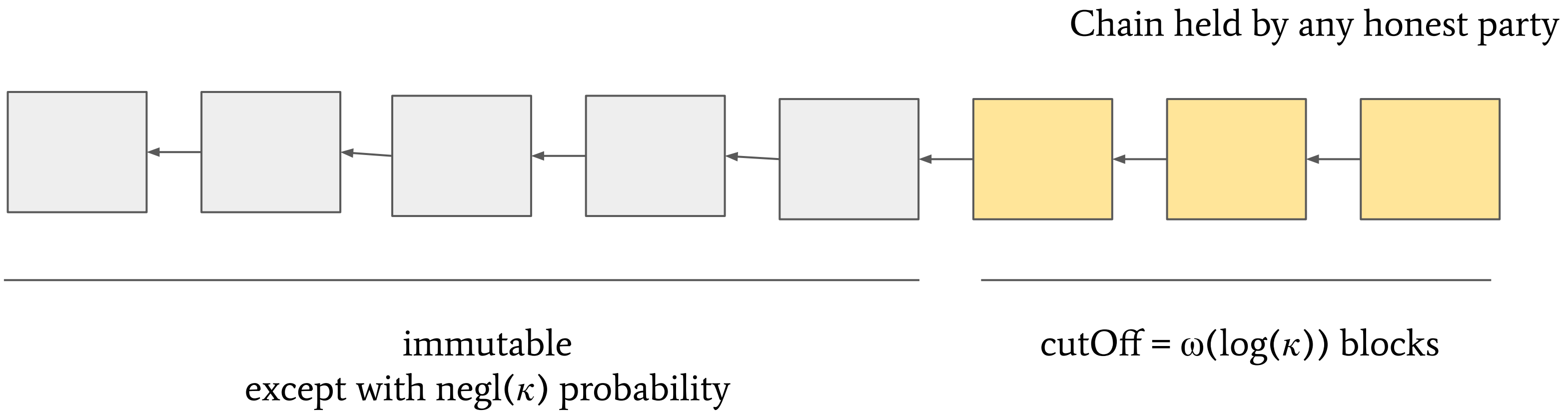
What happened to consistency?



Blockchain **consistency** is supposed to prevent double-spending!

- e.g. [Nakamoto 2008], [GKL 2015], [PSS 2017], [BMTZ 2017].... etc.

What happened to consistency?



Blockchain **consistency** is supposed to prevent double-spending!

- e.g. [Nakamoto 2008], [GKL 2015], [PSS 2017], [BMTZ 2017].... etc.

Breaking consistency

Two assumptions required for consistency:

- Bounded total hashing power
- ~~**Honest majority of hashing power**~~ (broken by 51% attacker)

Any attacker obtaining majority power (not just 51%)

When consistency is broken, we say there is a (deep) **fork** in the blockchain

Overview of Contributions

- Model 51% attacks in the rational protocol design framework (RPD)
- The problem of unbounded incentives
- What makes a coin susceptible to 51% attacks?
- How can we protect a coin from 51% attacks?

Overview of Contributions

- Model 51% attacks in the rational protocol design framework (RPD)
- The problem of unbounded incentives
- What makes a coin susceptible to 51% attacks?
- How can we protect a coin from 51% attacks?

51% attacks: Rational treatment

Q: Why are some blockchains more vulnerable to 51% attacks than others?

51% attacks: Rational treatment

Q: Why are some blockchains more vulnerable to 51% attacks than others?

A: Attackers care about **profit**! Factors to consider:

- Amount to be double-spent (e.g., 100 SadCoins)
- Cost to attack (e.g., cost to buying or renting mining rigs, electricity costs)
- Block rewards

51% attacks: Rational treatment

Q: Why are some blockchains more vulnerable to 51% attacks than others?

A: Attackers care about **profit**! Factors to consider:

- Amount to be double-spent (e.g., 100 SadCoins)
- Cost to attack (e.g., cost to buying or renting mining rigs, electricity costs)
- Block rewards

See also:

- [Bud18] - economics analysis; [JL20] - random walk; [GKW+16] and [HSY+21] - Markov Decision Process model
- Other rational analyses of blockchains e.g., [Ros11, CKWN16, ES14, Eya15, SBBR16, SSZ16, LTKS15, TJS16, NKMS16, PS17, GKW+16])

Rational protocol design (RPD) [GKMTZ13] (FOCS 2013)

Main advantages:

- Rational cryptographic model
- No restriction on adversary actions
- Composable

Rational protocol design (RPD)

Protocol Designer **D**

Blockchain
protocol **Π**

Rational protocol design (RPD)

Protocol Designer **D**

Wants to implement

Consistent ledger functionality **F**

Blockchain protocol **Π**

Rational protocol design (RPD)

Protocol Designer **D**

Wants to implement → Consistent ledger functionality **F**

Blockchain protocol **Π**

Can implement (because no honest majority) → Inconsistent ledger functionality **weak(F)** that allows blockchain forks

Rational protocol design (RPD)

Protocol Designer **D**

Wants to implement →

Consistent ledger functionality **F**

Blockchain protocol **Π**

→ Can implement (because no honest majority)

Inconsistent ledger functionality **weak(F)** that allows blockchain forks

Goal: Prove that we don't need the weaknesses in weak(F) to simulate a rational attacker (acting according to his utility function u_A)

Rational protocol design (RPD)

Protocol Designer **D**

Wants to implement

Consistent ledger functionality **F**

Blockchain protocol **Π**

Can implement (because no honest majority)

Inconsistent ledger functionality **weak(F)** that allo

Attack-payoff security

Goal: Prove that we don't need the weaknesses in weak(F) to simulate a rational attacker (acting according to his utility function u_A)

Rational protocol design (RPD)

[BGMTZ18] (Eurocrypt 2018):

Bitcoin backbone protocol has ***strong* attack-payoff security**

- Attack-payoff security: Rational attacker don't use weaknesses in weak(F).
- *Strong* attack-payoff security: Front-running, honest-mining is a dominant strategy

Rational protocol design (RPD)

[BGMTZ18] (Eurocrypt 2018):

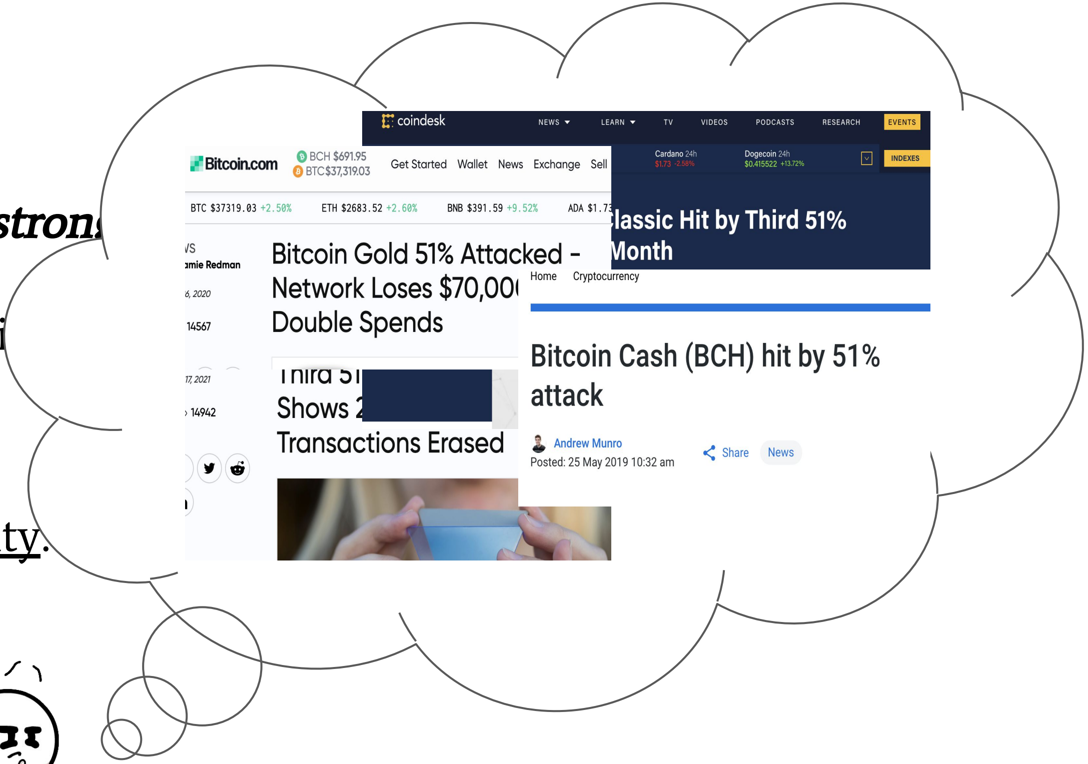
Bitcoin backbone protocol has ***stron***

- <u>Attack-payoff security</u>: Rati

  weak(F).

- <u>*Strong* attack-payoff security</u>.

  dominant strategy



Extending the utility with double-spending?

$$u_A(\Pi, A(\Pi))$$

$$\approx \sum_{(b, r)} b \cdot \text{breward} \cdot \Pr(I_{b,r}) - \sum_{(q, r)} q \cdot \text{mcost} \cdot \Pr(W_{q,r})$$

Extending the utility with double-spending?

$$u_A(\Pi, \ A(\Pi))$$

$$\approx \sum_{(b,\ r)} b \cdot \text{breward} \cdot \Pr(I_{b,r}) \ - \sum_{(q,\ r)} q \cdot \text{mcost} \cdot \Pr(W_{q,r})$$

Actually depends on the simulator in the ideal world, and the environment

Extending the utility with double-spending?

$$u_A(\Pi,\ A(\Pi))$$

$$\asymp \sum_{(b,\, r)} b \cdot \text{breward} \cdot \Pr(I_{b,r}) - \sum_{(q,\, r)} q \cdot \text{mcost} \cdot \Pr(W_{q,r})$$

Reward for making a block

Corrupt parties have b blocks confirmed in ledger at round r

and the environment

Extending the utility with double-spending?

$$u_A(\Pi, A(\Pi))$$

$$\approxeq \sum_{(b,r)} b \cdot \text{breward} \cdot \Pr(I_{b,r}) - \sum_{(q,r)} q \cdot \text{mcost} \cdot \Pr(W_{q,r})$$

Reward for making a block

Corrupt parties have b blocks confirmed in ledger at round r

and the environment

Cost of making one mining (hash) query

Make q queries in round r

Extending the utility with double-spending?

Reward for forking/breaking consistency (e.g. double-spend)

Probability of a fork

$$u_A(\Pi, \mathcal{A}(\Pi))$$

$$\approx \sum_{(b,\,r)} b \cdot \text{breward} \cdot \Pr(I_{b,r}) - \sum_{(q,\,r)} q \cdot \text{mcost} \cdot \Pr(W_{q,r}) + \text{fpayoff} \cdot \Pr(K)$$

Reward for making a block

Corrupt parties have b blocks confirmed in ledger at round r

and the environment

Cost of making one mining (hash) query

Make q queries in round r

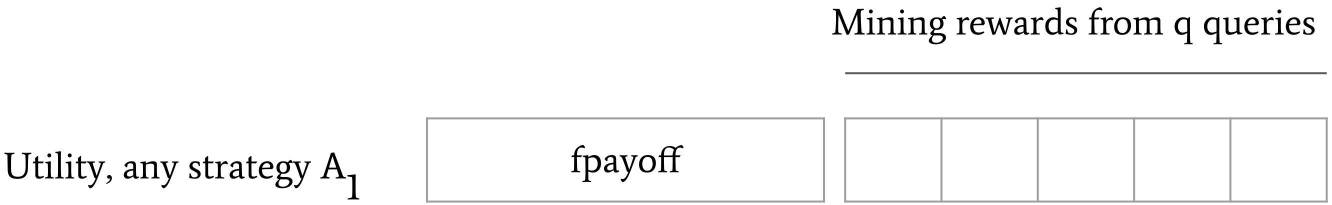
[BGMTZ18] => Still "secure"!

Lemma (informal): For arbitrarily large but poly-size fpayoff (e.g., payoff for double-spending), blockchain is strongly attack payoff secure.

[BGMTZ18] => Still "secure"!

Lemma (informal): For arbitrarily large but poly-size fpayoff (e.g., payoff for double-spending), blockchain is strongly attack payoff secure.

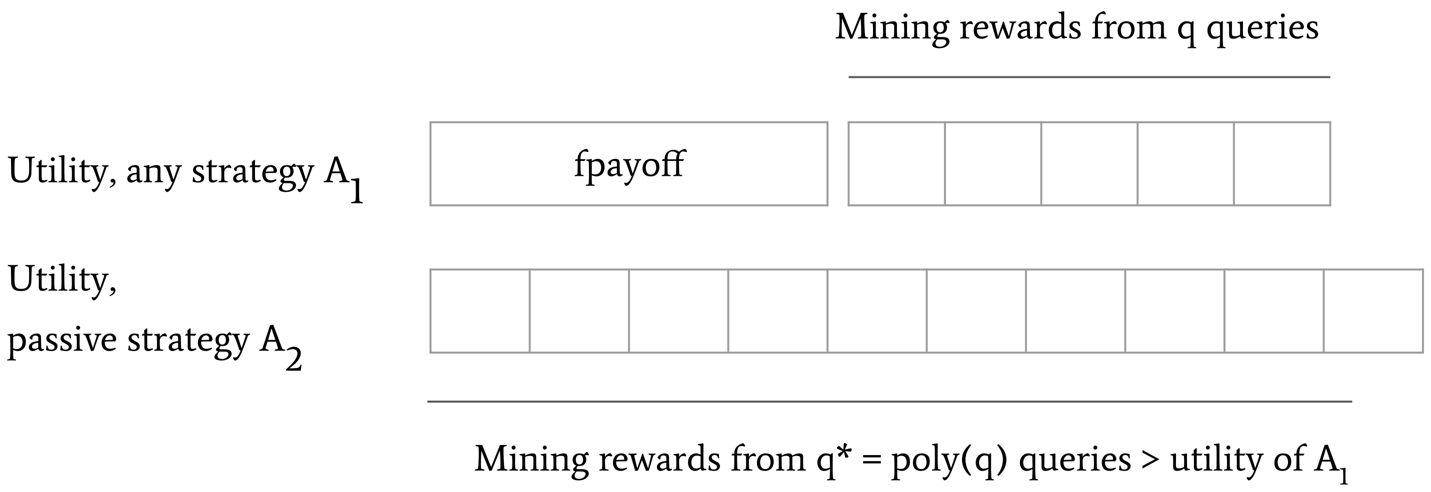
Proof (similar to [BGMTZ18]):



[BGMTZ18] => Still "secure"!

<u>Lemma (informal)</u>: For arbitrarily large but poly-size fpayoff (e.g., payoff for double-spending), blockchain is strongly attack payoff secure.

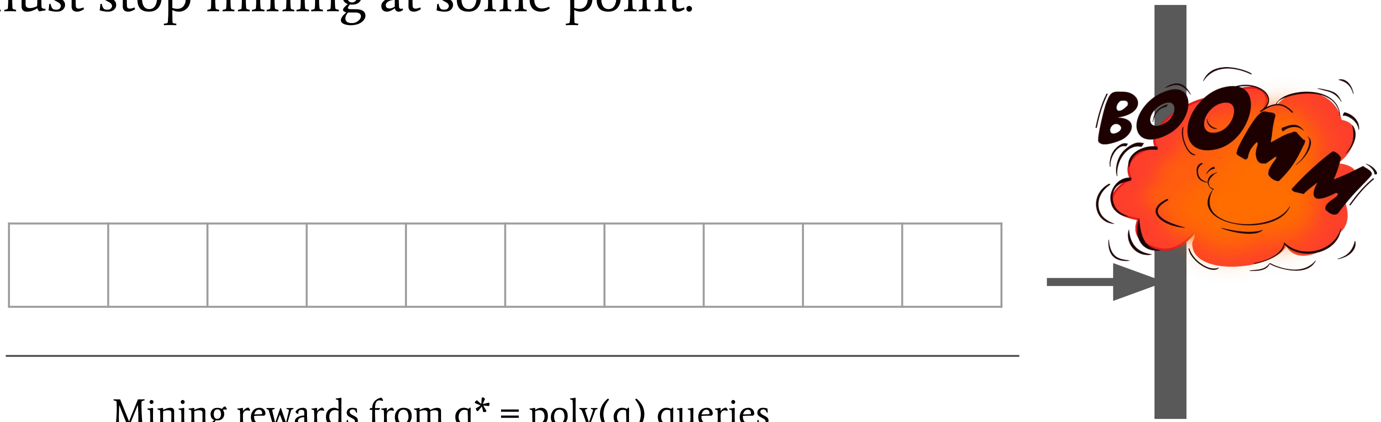
<u>Proof (similar to [BGMTZ18])</u>:



Problem

Realistically, one must stop mining at some point.

Utility,
passive strategy A_2



Mining rewards from $q^* = poly(q)$ queries

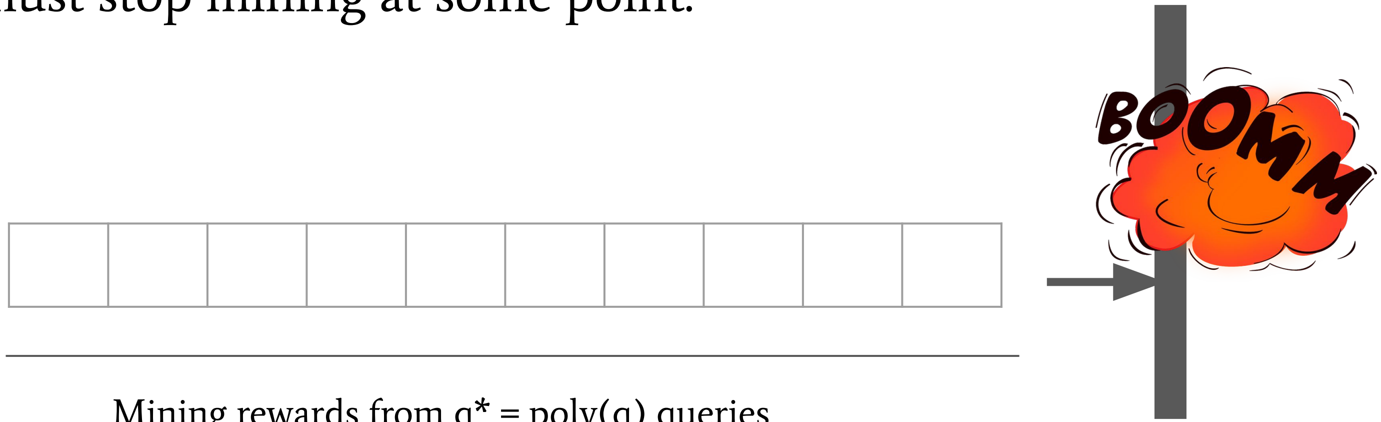
e.g., Estimated End of the Universe

Problem

Realistically, one must stop mining at some point.

Utility,

passive strategy A_2



Mining rewards from q* = poly(q) queries

e.g., Estimated End of the Universe

- Cannot amplify amount of passive mining rewards forever
- Example of *St. Petersburg paradox*

Overview of Contributions

- Model 51% attacks in the rational protocol design framework (RPD)
- The problem of unbounded incentives
- What makes a coin susceptible to 51% attacks?
- How can we protect a coin from 51% attacks?

Unbounded incentives

“Unbounded incentives”:

Utility functions with unlimited growth of utility for passive adversaries.

Lemma (informal):

Any protocol (no matter how “good” or “bad” it is!) is strongly-attack payoff secure, if the attacker’s utility function has unbounded incentives.

Limited horizons: avoiding “unbounded incentives”

$$u_A(\Pi, A(\Pi))$$

$$\approxeq \sum_{(b,r)} b \cdot \text{breward}(r) \cdot \Pr(I_{b,r}) - \sum_{(q,r)} q \cdot \text{mcost} \cdot \Pr(W_{q,r}) + \text{fpayoff} \cdot \Pr(K)$$

- $u_A(\Pi, A(\Pi))$ has **limited horizons** if breward(r) is a non-increasing function and there is a round **r** such that after r:

E(block reward at round r) – E(mining costs at round r) < 0

- Easy to see limited horizons utility -> NOT unbounded

Overview of Contributions

- 51% attacks in the rational protocol design model (RPD)
- The problem of unbounded incentives
- What makes a coin susceptible to 51% attacks?
- How can we protect a coin from 51% attacks?

What makes a coin susceptible to 51% attacks?

Theorem: (Very roughly) For limited horizons utility function u_A, both attack-payoff security and strong attack-payoff security are impossible if

Lower bound utility of forking adversary	>	Upper bound utility of optimal front-running, passive-mining adversary

Upper bound optimal passive-mining utility

$$u_A(\Pi, A(\Pi))$$

$$\approx \sum_{(b,r)} b \cdot \text{breward}(r) \cdot \Pr(I_{b,r}) - \sum_{(q,r)} q \cdot \text{mcost} \cdot \Pr(W_{q,r}) + \text{fpayoff} \cdot \Pr(K)$$

Upper bound optimal passive-mining utility

$$u_A(\Pi,\ A(\Pi))$$

$$\lessapprox \sum_{(b,\,r)} b \cdot \mathrm{breward}(r) \cdot \Pr(I_{b,r}) - \sum_{(q,\,r)} q \cdot \mathrm{mcost} \cdot \Pr(W_{q,r}) \; \cancel{+\ \mathrm{fpayoff} \cdot \Pr(K)}$$

Main observations:

1. $\Pr(K) = \mathrm{negl}(\kappa)$

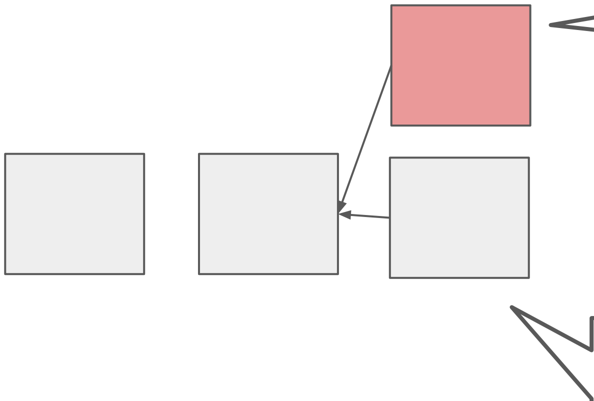
Upper bound optimal passive-mining utility

$$u_A(\Pi,\ A(\Pi)) \approx \boxed{\sum_{(b,\,r)} b \cdot \mathrm{breward}(r) \cdot \Pr(I_{b,r})} - \sum_{(q,\,r)} q \cdot \mathrm{mcost} \cdot \Pr(W_{q,r}) \cancel{+ \mathrm{fpayoff} \cdot \Pr(K)}$$

Main observations:

1. $\Pr(K) = \mathrm{negl}(\kappa)$
2. The term $\boxed{\sum_{(b,\,r)} b \cdot \mathrm{breward}(r) \cdot \Pr(I_{b,r})}$ is hard to compute (time of block enters the ledger = hard to predict) but can be upper-bounded by using time of block broadcast.

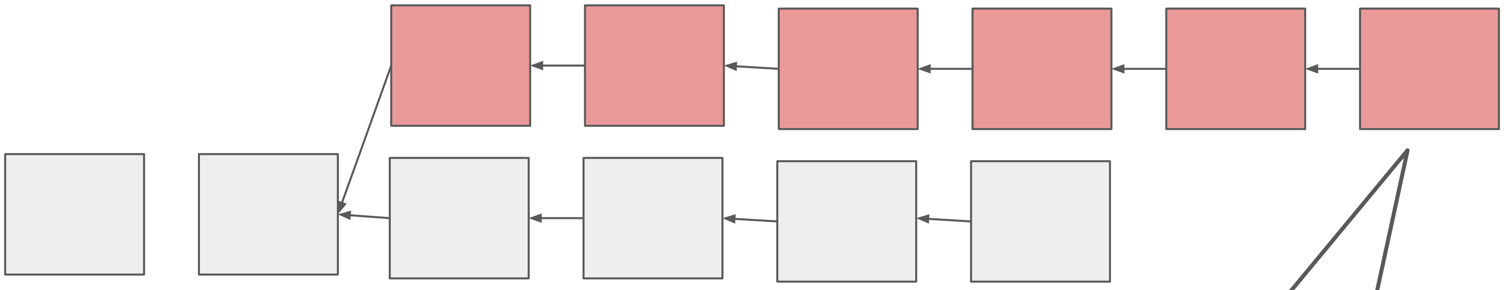
Lower bound utility of forking adversary



Adversary only works on this chain and keeps successful blocks private

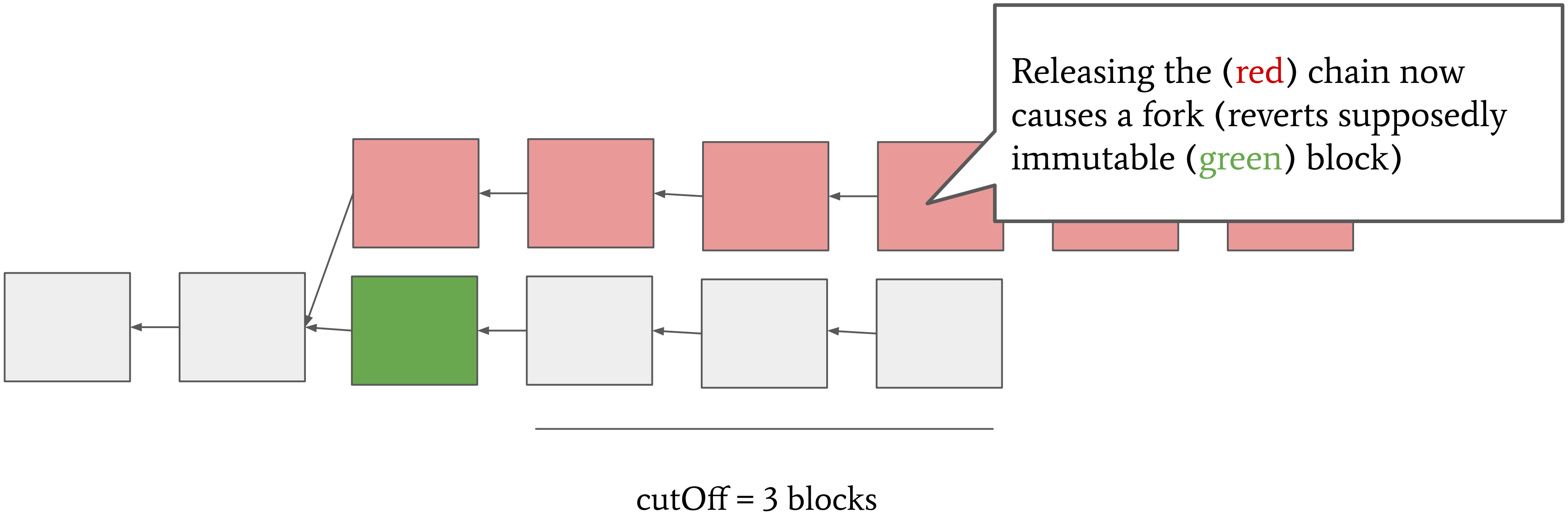
Honest parties continue with this chain

Lower bound utility of forking adversary

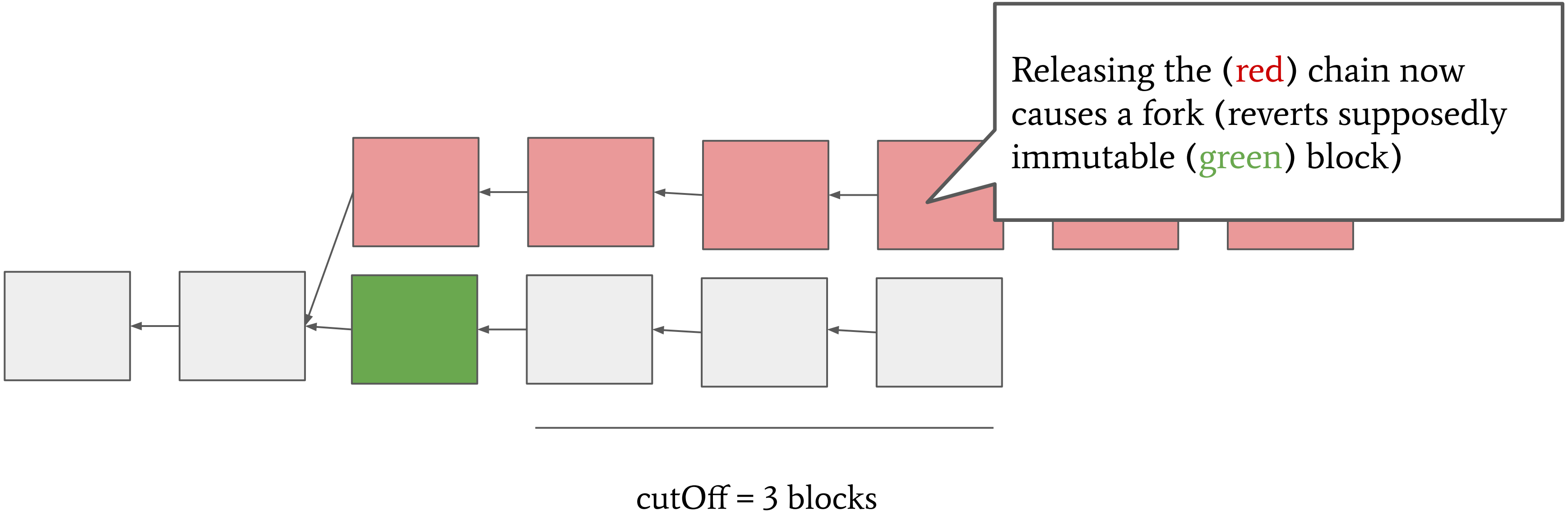


Adversary has majority hashing power, so his privately-kept chain grows faster

Lower bound utility of forking adversary



Lower bound utility of forking adversary



How long this takes depends on growth speed of lower chain -- ***Chain growth***

Lower bound utility of forking adversary

Let t_q = time it takes until a fork is possible using this adversarial strategy

$$u_A \geq E(\text{Block rewards} - \text{mining costs in } t_q) + \text{fpayoff}$$

Adversary forks with overwhelming probability

Overview of Contributions

- Model 51% attacks in the rational protocol design framework (RPD)
- The problem of unbounded incentives
- What makes a coin susceptible to 51% attacks?
- How can we protect a coin from 51% attacks?

How to protect coins from 51% attacks?

- **No** restriction on adversarial strategy
- **No** assumption of honest majority, only that attackers are rational

How to protect coins from 51% attacks?

- **No** restriction on adversarial strategy
- **No** assumption of honest majority, only that attackers are rational

Q: How much confirmation time for a block to be immutable in the blockchain?

Modeling and restricting 51% attackers

We say an adversary **spends budget B** [BGKRZ20] if he makes a total of B mining queries over majority of total hashing power.

- e.g. (very informally) if the total hashing power in the system is 100 mining queries/round, and he makes $51 = 50\% \times 100 + 1$ queries in one round, he spent budget $B = 1$ in this round.

Modeling and restricting 51% attackers

We say an adversary **spends budget B** [BGKRZ20] if he makes a total of B mining queries over majority of total hashing power.

- e.g. (very informally) if the total hashing power in the system is 100 mining queries/round, and he makes $51 = 50\% \times 100 + 1$ queries in one round, he spent budget $B = 1$ in this round.

<u>Proof idea:</u>

Limited horizons utility function u_A

Modeling and restricting 51% attackers

We say an adversary **spends budget B** [BGKRZ20] if he makes a total of B mining queries over majority of total hashing power.

- e.g. (very informally) if the total hashing power in the system is 100 mining queries/round, and he makes $51 = 50\% \times 100 + 1$ queries in one round, he spent budget $B = 1$ in this round.

Proof idea:

Upper bound utility u(B, t)
of adversary spending B budget over t rounds

Limited horizons utility function u_A

Modeling and restricting 51% attackers

We say an adversary **spends budget B** [BGKRZ20] if he makes a total of B mining queries over majority of total hashing power.

- e.g. (very informally) if the total hashing power in the system is 100 mining queries/round, and he makes 51 = 50% × 100 + 1 queries in one round, he spent budget B = 1 in this round.

<u>Proof idea:</u>

Limited horizons utility function u_A

→ **Upper bound utility u(B, t)** of adversary spending B budget over t rounds

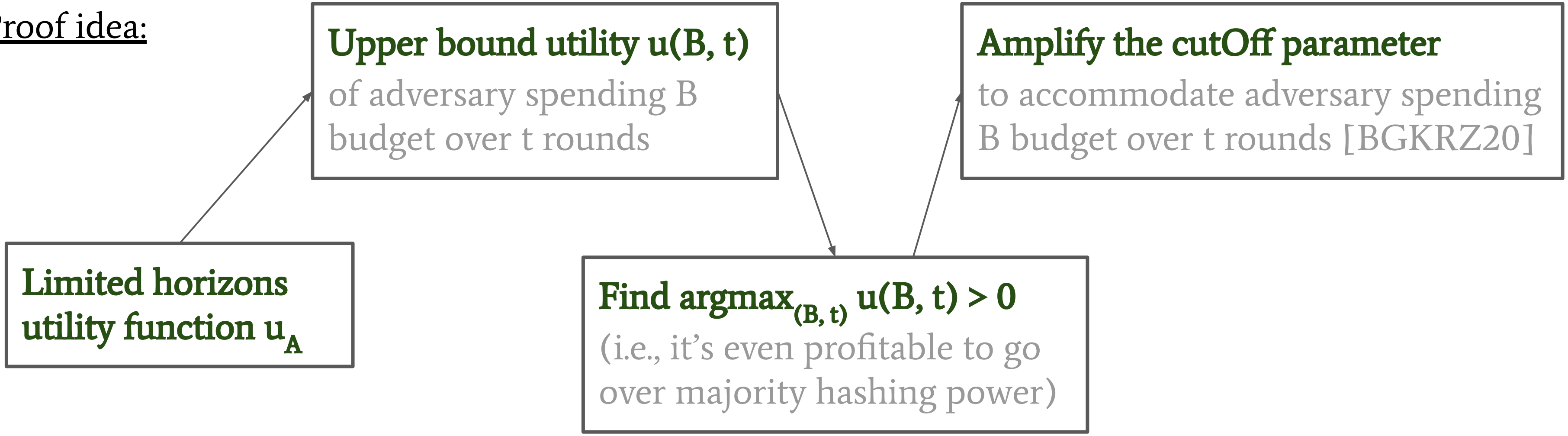
→ **Find $\text{argmax}_{(B,t)}$ u(B, t) > 0** (i.e., it's even profitable to go over majority hashing power)

Modeling and restricting 51% attackers

We say an adversary **spends budget B** [BGKRZ20] if he makes a total of B mining queries over majority of total hashing power.

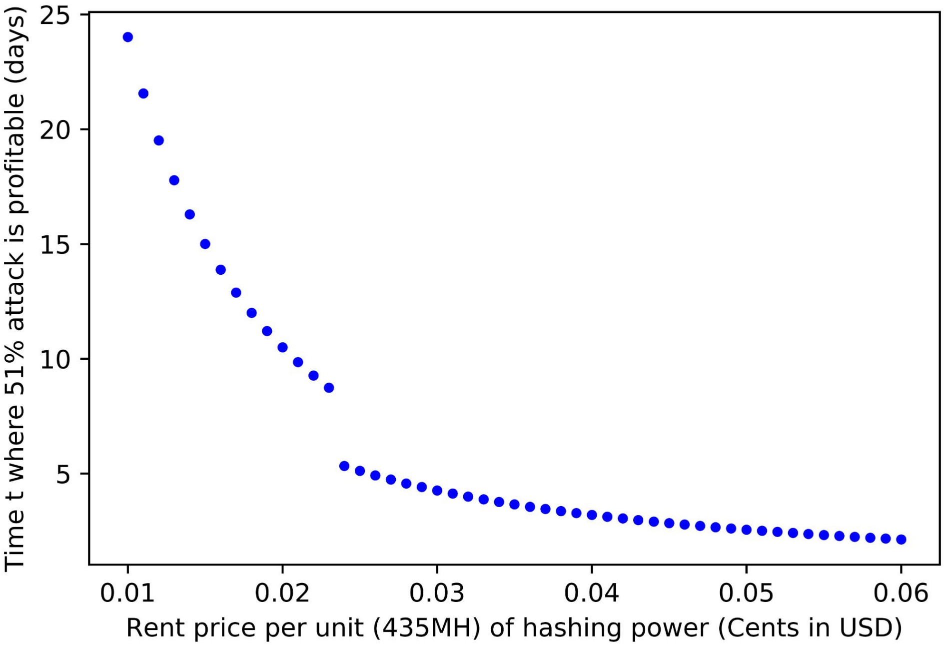
- e.g. (very informally) if the total hashing power in the system is 100 mining queries/round, and he makes 51 = 50% × 100 + 1 queries in one round, he spent budget B = 1 in this round.

<u>Proof idea:</u>



Visualizing 51% attacks for Ethereum Classic

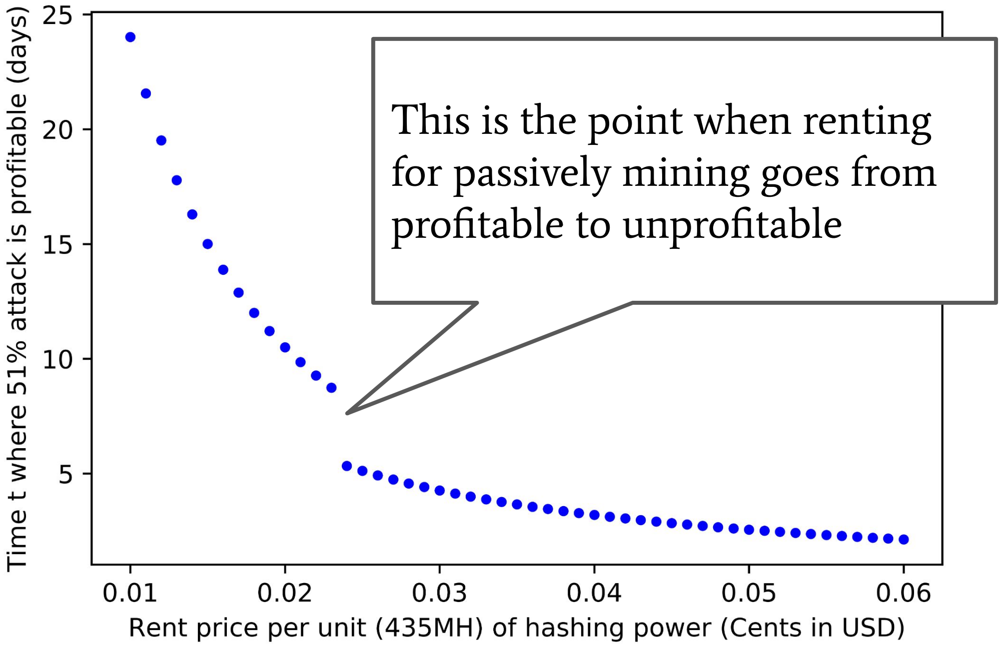
USD cost of renting 435MH/s	Length of attack* (argmax$_t$ u(· , t) > 0)
$ 0.0001	24 days
$ 0.0002	10.5 days
$ 0.0003	4.3 days
$ 0.0004	3.2 days
$ 0.0005	2.6 days
$ 0.0006	2.1 days



* Using parameters for Ethereum Classic from Feb, 2021. Using t = 3 days as max interval where passive mining is on expectation profitable (for limited horizons).

Visualizing 51% attacks for Ethereum Classic

USD cost of renting 435MH/s	Length of attack* ($\text{argmax}_t\ u(\cdot, t) > 0$)
$ 0.0001	24 days
$ 0.0002	10.5 days
$ 0.0003	4.3 days
$ 0.0004	3.2 days
$ 0.0005	2.6 days
$ 0.0006	2.1 days



* Using parameters for Ethereum Classic from Feb, 2021. Using t = 3 days as max interval where passive mining is on expectation profitable (for limited horizons).

Summary

- Realistic utility functions must avoid unbounded incentives
- Limited horizons utility functions analyses both
  1. When attack-payoff security is broken (forking is profitable over honestly-mining)
  2. When attack-payoff security is maintained

Future work:

- Practical implementations
- Analyzing more complex utility functions
- Analyzing variable difficulty blockchain

  (e.g., extending from analyses of [GKL20], [CEMMPS20])

References

[Nakamoto2008] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[GKL2015] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. EUROCRYPT 2015

[PSS2017] Rafael Pass, Lior Seeman, and abhi shelat. Analysis of the blockchain protocol in asynchronous networks. EUROCRYPT 2017

[BMTZ2017] Christian Badertscher, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. Bitcoin as a transaction ledger: A composable treatment. CRYPTO 2017

[Bud18] Eric Budish. The economic limits of bitcoin and the blockchain. Technical report, National Bureau of Economic Research, 2018.

[JL20] Jehyuk Jang and Heung-No Lee. Profitable double-spending attacks. Applied Sciences

[GKW+16] Arthur Gervais, Ghassan O. Karame, Karl Wust, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. CCS 2016

References

[HSY+21] Runchao Han, Zhimei Sui, Jiangshan Yu, Joseph Liu, and Shiping Chen. Fact and fiction: Challenging the honest majority assumption of permissionless blockchains. ASIA CCS 2021

[Ros11] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. CoRR, 2011.

[CKWN16] Miles Carlsten, Harry A. Kalodner, S. Matthew Weinberg, and Arvind Narayanan. On the instability of bitcoin without the block reward. CCS 2016.

[ES14] Ittay Eyal and Emin G un Sirer. Majority is not enough: Bitcoin mining is vulnerable. In Nicolas Christin and Reihaneh Safavi-Naini, editors, FC 2014.

[Eya15] Ittay Eyal. The miner's dilemma. Security and Privacy 2015

[SBBR16] Okke Schrijvers, Joseph Bonneau, Dan Boneh, and Tim Roughgarden. Incentive compatibility of bitcoin mining pool reward functions. FC 2016

References

[SSZ16] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. FC 2016

[LTKS15] Loi Luu, Jason Teutsch, Raghav Kulkarni, and Prateek Saxena. Demystifying incentives in the consensus computer. CCS 2015

[TJS16] Jason Teutsch, Sanjay Jain, and Prateek Saxena. When cryptocurrencies mine their own business. FC 2016,

[NKMS16] Kartik Nayak, Srijan Kumar, Andrew Miller, and Elaine Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In S&P, 2016

[PS17] Rafael Pass and Elaine Shi. FruitChains: A fair blockchain. PODC 2017

[GKMTZ13] Juan A. Garay, Jonathan Katz, Ueli Maurer, Bj orn Tackmann, and Vassilis Zikas. Rational protocol design: Cryptography against incentive driven adversaries. FOCS 2013.

[BGMTZ18] Christian Badertscher, Juan A. Garay, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. But why does it work? A rational protocol design treatment of bitcoin. EUROCRYPT 2018,

References

[BGKRZ20] Christian Badertscher, Peter Ga?i, Aggelos Kiayias, Alexander Russell, and Vassilis Zikas. Consensus redux: Distributed ledgers in the face of adversarial supremacy. Cryptology ePrint Archive

[GKL20] Juan Garay and Aggelos Kiayias and Nikos Leonardos. Full Analysis of Nakamoto Consensus in Bounded-Delay Networks. Cryptology ePrint Archive

[CEMMPS20] T-H. Hubert Chan and Naomi Ephraim and Antonio Marcedone and Andrew Morgan and Rafael Pass and Elaine Shi. Blockchain with Varying Number of Players. Cryptology ePrint Archive

Thanks for watching!