SBA Research

TU WIEN TECHNISCHE UNIVERSITÄT WIEN Vienna University of Technology

Imperial College London

TECHNION Israel Institute of Technology

INPUT | OUTPUT

UCL

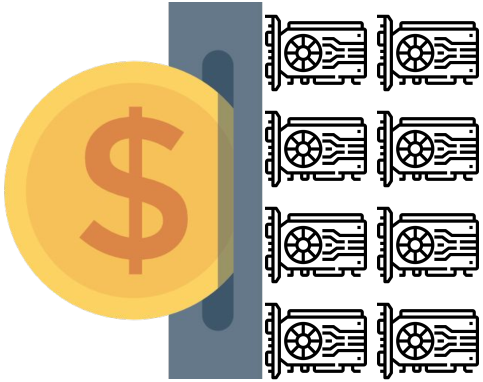
IC3 The Initiative For CryptoCurrencies & Contracts

*Devcon 5 Research Meetup*

*11 October 2019, Osaka, Japan*

# Pay-to-Win

## Incentive Attacks on Proof-of-Work Cryptocurrencies




*Aljosha Judmayer, Nicholas Stifter, Alexei Zamyatin,*

*Itay Tsabary, Ittay Eyal, Peter Gazi,*

*Sarah Meiklejohn, and Edgar Weippl*

*“The system is secure as long as **honest** nodes collectively control more CPU power than any cooperating group of attacker nodes.”*

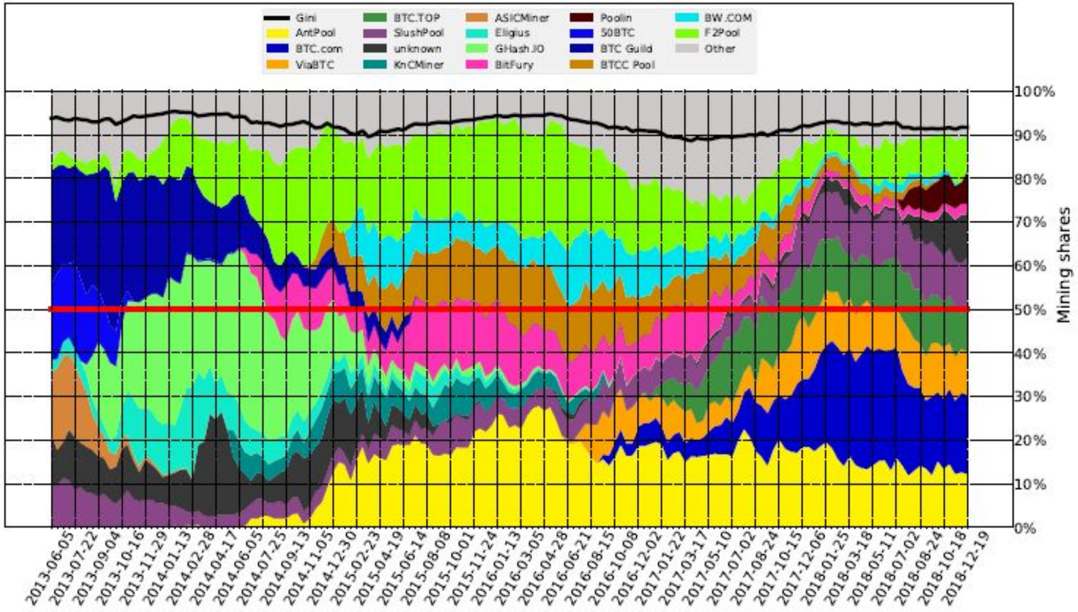
*Satoshi Nakamoto*

# Bitcoin's Security Model

… relies on 2/3 of the computational power being honest

But can we even determine if this is the case?

- Miners can collude
- Can be same entity
- …




**A Deep Dive into Bitcoin Mining Pools: An Empirical Analysis of Mining Shares**. Romiti M, Judmayer A, Zamyatin A, Haselhofer B. *Workshop on the Economics of Information Security (WEIS)*, 2019

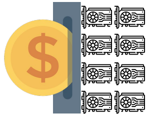
# BAR Model

Instead of only honest / dishonest actors, BAR model assumes:

- **Byzantine:** our adversary, behaves dishonestly
- **Altruistic:** altruistic motives, behave honestly
- **Rational:** may deviate from rules to maximize profit

→ **Bribing attacks** assume economically **rational** actors can be bribed into misbehaving

# “*Why buy when you can rent?*”



**Idea of Bribing attacks:**

- Attacker does not need to be a miner
- Offers payment to miners to attack underlying chain
- Ideally: miners do not have to trust the adversary
  - e.g. via smart contracts

**Goals:**

- Censorship, double spending, reducing active hash rate, destruction on the coin, ...

# Recall: States of a Transaction

**Unconfirmed**

TX has been **broadcast** to the network.
(*“proposed” or “published”*)

**Confirmed**

TX has been **included** in a block

**Agreed**

TX has been agreed upon, i.e., has ***consensus***

→ it has received **k** confirmations and revision is highly unlikely

k - security parameter dependent on underlying chain*

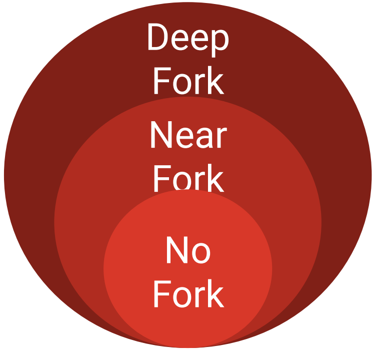
* More about this later

# Impact and Required Interference

## Impact on Transactions

| | |
|---|---|
| **Revision** | Change published, confirmed or agreed TX |
| **Re-ordering** | Change ordering of published, confirmed or agreed TX in a block |
| **Exclusion / Censorship** | Prevent TX from from being included in the chain (for some period) |

## Interference with Consensus




- **Deep forks**
  Exceeding the security parameter **k selected by the victim**
- **Near forks**
  Fork, but depth is **not dependent on victim's k parameter**
- **No forks**

# Further Properties

1. Required **attacker hash rate**
2. Required **rational miner hash rate**
3. **Distract hash rate**?
4. **Smart contracts** required?
5. Must the **attacker trust miners**?
6. Must **miners trust the attacker**?
7. Are **failed attacks compensated**?
8. Coordination / payment **in-band or out-of-band (cross-chain)**?
9. …

See paper for more details!

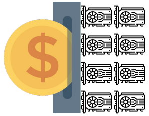
# Classification of Incentive Attacks

| | Tx rev. | Tx ord. | Tx excl. | Required chain reorganization | Attacker hashrate $\alpha$ | Rational hashrate $\omega$ | Distracts hashrate | Requires smart contract | Payment | Trustless for attacker | Trustless for collaborator | Subsidy | Compensates if attack fails |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Checklocktime bribes [7] | ✓ | ✗ | ✗ | Deep fork | ✗ | $\approx [\frac{1}{2}, 1]$ | ✗ | ✗ | in-band | ✓ | $\sim$ | ✗ | ✗ |
| Whale Transactions [19] | ✓ | ✗ | ✗ | Deep fork | ✗ | $\approx [\frac{1}{2}, 1]$ | ✗ | ✗ | in-band | ✓ | $\sim$ | ✗ | ✗ |
| Script Puzzle double-spend [30] | ✓ | $\sim$ | ✓ | Deep fork | $(0, \frac{1}{2})$ | $1 - \alpha$ | ✓ | ✗ | in-band | $\sim$ | ✗ | ✗ | $\sim$ |
| Script Puzzle 38.2% attack [30] | ✗ | $\sim$ | ✓ | Near-/No forks | $[0.382, \frac{1}{2})$ | $1 - \alpha$ | ✓ | ?† | out-of-band | ?† | ?† | ✗ | ✓ |
| Proof-of-Stale blocks [20], [32] | -⋆ | -⋆ | -⋆ | -⋆ | ✗ | - | ✓ | ✓ | out-of-band | $\sim$ | ✓ | ✗ | ✓ |
| CensorshipCon [21] | ✗ | $\sim$ | ✓ | Near-/No forks | $[\frac{1}{3}, \frac{1}{2})$ | $[\frac{1}{3}, \frac{2}{3})$ | ✓ | ✓ | in-band | $\sim$ | ✗ | ✓ | ✗ |
| HistoryRevisionCon [21] | ✓ | ✗ | ✗ | Deep fork | ✗ | $\approx [\frac{1}{2}, 1]$ | ✗ | ✓ | in-band | ✓ | $\sim$ | ✓ | ✗ |
| GoldfingerCon [21] | - | - | ✓all | No fork | ✗ | $\approx [\frac{1}{2}, 1]$ | ✗ | ✓ | out-of-band | ✓ | ✓ | ✗ | ✓ |
| Pitchforks [15] | - | - | ✓all | No fork | ✗ | $(\frac{1}{3}, 1]$ | ✓ | ✗ | out-of-band | ✓ | ✓ | ✓ | ✗ |
| Front-running [10], [12] | ✗ | ✓ | ✗ | No fork | ✗ | $(0, 1]$ | ✗ | ✗ | in-band | ✗ | ✓ | ✗ | ✓ |
| Pay per Miner Censorship [33] | ✗ | ✗ | ✓ | No fork | ✗ | 1 | ✗ | ✓ | in-band | ✓ | ✓ | ✗ | ✗ |
| Pay per Block Censorship [33] | ✗ | ✗ | ✓ | No fork | ✗ | 1 | ✗ | ✓ | in-band | ✓ | ✓ | ✗ | ✓ |
| Pay per Commit Censorship [33] | ✗ | ✗ | ✓ | Near-/No fork | ✗ | 1 | ✗ | ✓ | in-band | ✓ | ✓ | ✗ | ✗ |
| P2W Tx Excl.& Ord. | ✗ | ✓ | ✓ | Near-/No forks | ✗ | $[\frac{1}{2}, 1]$ | ✗ | ✓ | out-of-band | ✓ | ✓ | ✗ | ✓ |
| P2W Tx Rev. & Excl. & Ord. | ✓ | ✓ | ✓ | Deep fork | ✗ | $[\frac{1}{2}, 1]$ | ✗ | ✓ | out-of-band | ✓ | ✓ | ✗ | ✓ |
| P2W Tx Ord. Appendix E | ✗ | ✓ | ✗ | No fork | ✗ | $(0, 1]$ | ✗ | ✓ | in-band | ✓ | ✓ | ✗ | ✗ |
| P2W Tx Excl. Appendix F | ✗ | ✗ | ✓ | Near-/No forks | ✗ | $[\frac{1}{2}, 1]$ | ✗ | ✓ | in-band | ✓ | ✓ | ✗ | ✗ |

See paper for more details!

# Bribing Myths

# *“Pfff, bribing is too expensive anyway…”*



Risk of failure must be compensated

**Existing bribing attacks**:
- Payment only if attack succeeds
- Overcompensate risk via **high bribes**

# ~~“Pfff, bribing is too expensive anyway…”~~

Risk of failure must be compensated

**Existing bribing attacks**:

- Payment only if attack succeeds
- Overcompensate risk via **high bribes**

**Pay-to-Win (This work):**

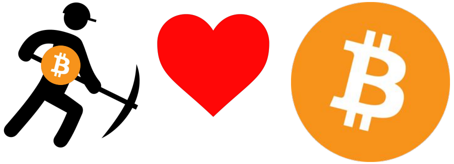
- Always pay miners, even if attack fails
- Miners face no financial risk

→ **only small bribes required**


**cheaper than existing attacks**

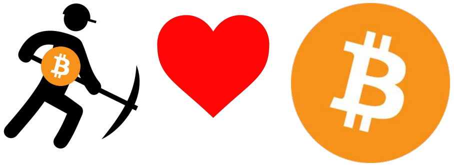
# “But miners will not attack their own coin!”

- One of the oldest arguments in this space
- Assumes miners have long term stake in their system



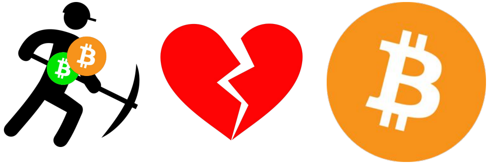
# *“But miners will not attack their own coin!”*

- One of the oldest arguments in this space
- Assumes miners have long term stake in their system



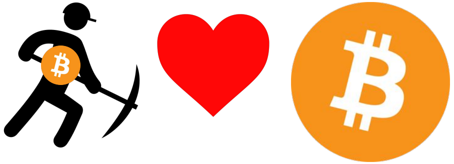
**Does not consider**:

- Private information



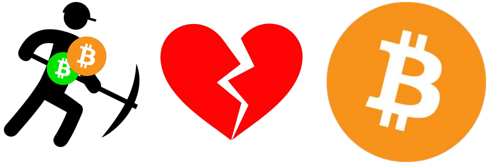
# ~~“But miners will not attack their own coin!”~~

- One of the oldest arguments in this space
- Assumes miners have long term stake in their system



**Does not consider**:

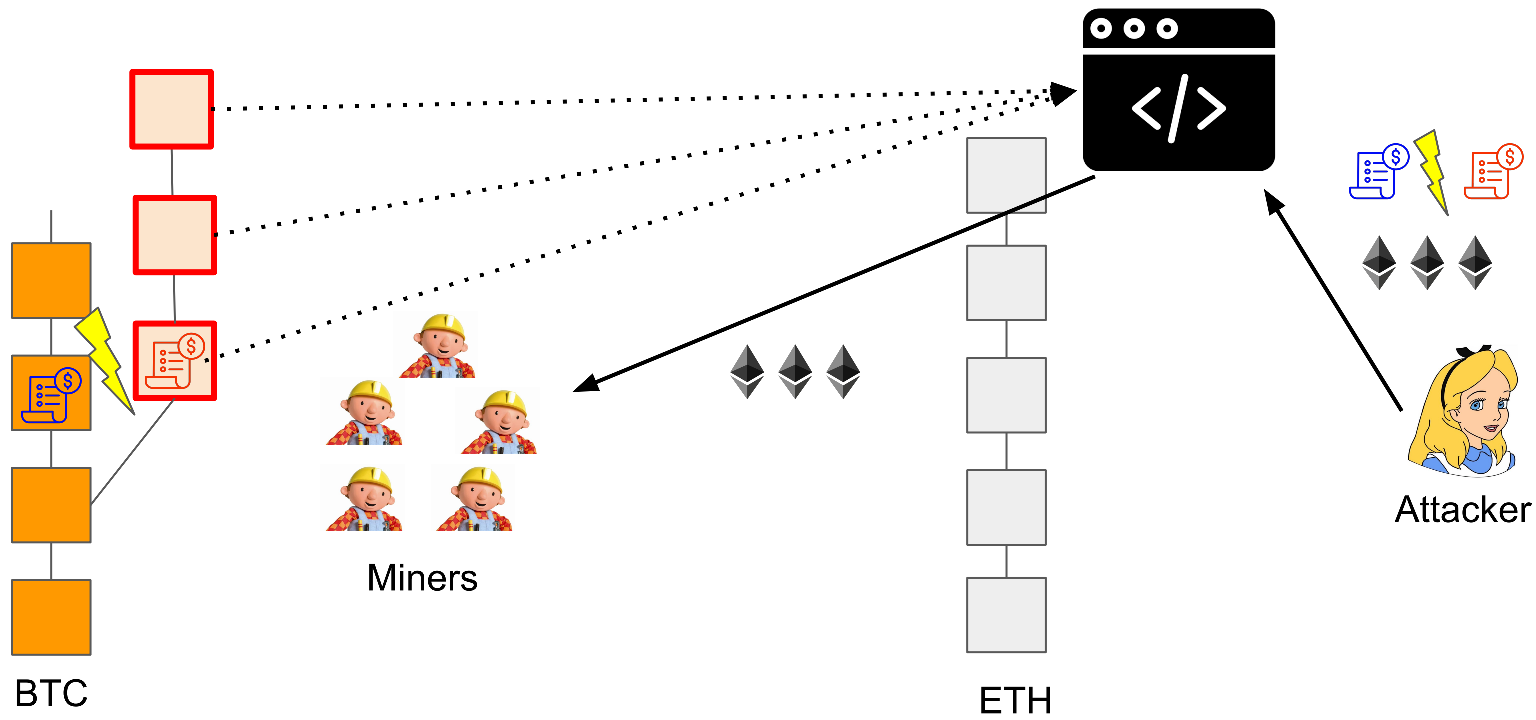
- Private information



- Cross-chain (“out of band”) attacks **(This work)**

# Cross-Chain Bribing Attacks

- **Coordination** and **payout** occur on **another chain**



# Cross-Chain Bribing Attacks

- **Coordination** and **payout** occur on **another chain**

  → ***Ephemeral mining relays*** **(This work)**

  1. **Verify state agreement & evolution** of target chain
  2. **Check validity** of blocks (pre-defined block & TX templates)
  3. **Track forks**
  4. **Check correct execution** of attack
  5. **Handle payouts** depending on outcome

# *“But is this not too complex and inefficient?”*

- **PoW verification needs to be supported by the funding chain!**

# ~~“But is this not too complex and expensive?”~~

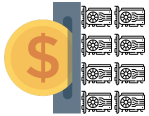
- **PoW verification needs to be supported by the funding chain!**
- PoC implementation of components for attacks on BTC, coordinated on ETH

Exaggerated example: 24h attack on Bitcoin (144 blocks)

- Costs to run relay:
  **~ 10-23 USD**
- For comparison:
  Value of single BTC block (excl. TX fees):
  **~ 77 000 USD**

| **Operation** | **Approx. costs** | |
|---|---|---|
| | **Gas** | **USD** |
| *Initialization* | 244 137 | 0.21 |
| *Block parsing and verification* | 174 929 | 0.15 |
| *Block header storage* | 141 534 | 0.12 |
| *Transaction parsing* | 117 253 | 0.1 |
| *Markle tree verification* | 80 257 - 194 351 | 0.07 - 0.16 |

Gas price: 5 Gwei, Exchange rates as per 10 May 2019 (168.01 USD/ETH)



# Pay-to-Win Attacks

# Overview

- Coordination and payouts happen **out-of-band (cross-chain)**
  - *Target* chain (e.g. Bitcoin) vs *funding* chain (e.g. Ethereum)

- Miners are **always compensated** (even for failed attacks)

- Uses **smart contracts on funding chain**
  → **trustless** for attacker and miners!

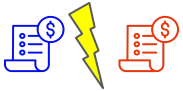
- **2 Variants:**
  - **No / near fork**: ordering and exclusion/censorship
  - **Deep fork**: revision, ordering and exclusion/censorship

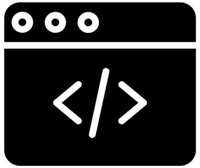
# Pay-to-Win Attack (Deep Fork)

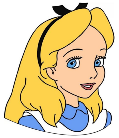
**Example**:  double spend on BTC

Attack suceeds if:

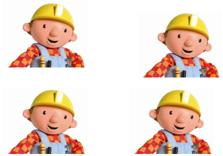
- **> k** blocks on main chain
- **> k+1** blocks on attack chain







Attacker



BTC

Miners

ETH

# Pay-to-Win Attack (Deep Fork)

**Example**: double spend on BTC
Attack suceeds if:

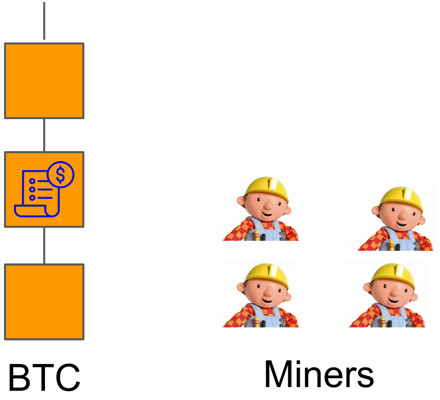
- **> k** blocks on main chain
- **> k+1** blocks on attack chain

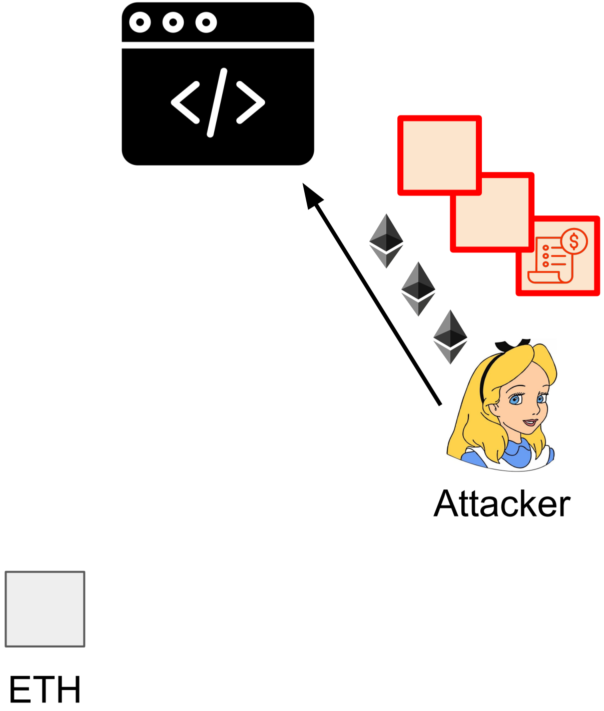
Attacker waits until victim’s TX is included and has **k** confirmations (**k** defined by victim)

# Pay-to-Win Attack (Deep Fork)

**Initialization Phase**:
Attacker initializes contract with

- *block templates*→ contain conditions for attack
- *compensation*



# Block Templates

Miners can only freely choose:

- *nonce* … for mining iteration
- *coinbase* ... link Ethereum account to block

Block Header

| |
|---|
| Version |
| PrevBlockHash |
| MerkleRoot |
| Time |
| nBits |
| nonce |

Coinbase TX

| | |
|---|---|
| nVersion | |
| #vin = 1 | |
| vin[0] | hash |
| | n |
| | coinbaseLen |
| | coinbase |
| | nSequence |
| #vout = 1 | |
| vout[0] | nValue |
| | scriptPubkeyLen |
| | scriptPubkey |
| nLockTime | |

# Block Templates

Miners can only freely choose:

- *nonce* … for mining iteration
- *coinbase* ... link Ethereum account to block

Block Header

| |
|---|
| Version |
| PrevBlockHash |
| MerkleRoot |
| Time |
| nBits |
| nonce |

Coinbase TX

| | |
|---|---|
| nVersion | |
| #vin = 1 | |
| vin[0] | hash |
| | n |
| | coinbaseLen |
| | coinbase |
| | nSequence |
| #vout = 1 | |
| vout[0] | nValue |
| | scriptPubkeyLen |
| | scriptPubkey |
| nLockTime | |

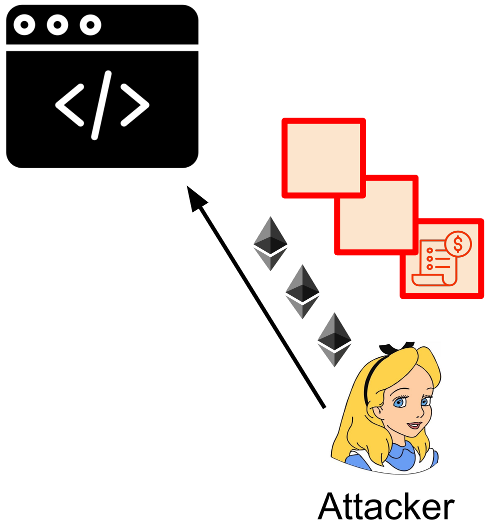
Note: **BTC block reward must go to attacker**
→ block reward compensation after
the attack ends in ETH

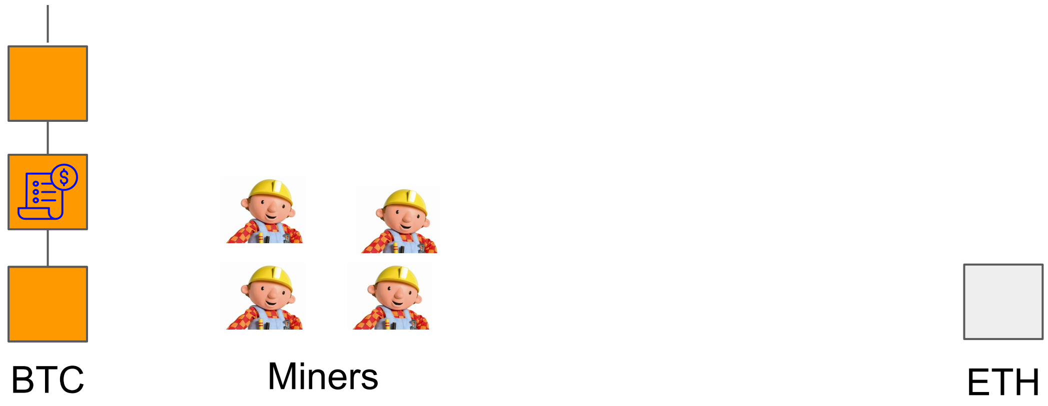
# Pay-to-Win Attack (Deep Fork)

**Initialization Phase**:
Attacker initializes contract with

- *block templates*→ contain conditions for attack
- *compensation*

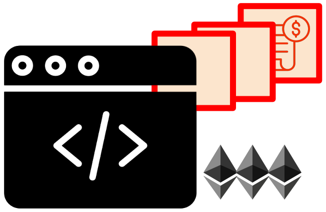
Once initialized: **no abort**! (or very high timelock)
→ Reason: race conditions



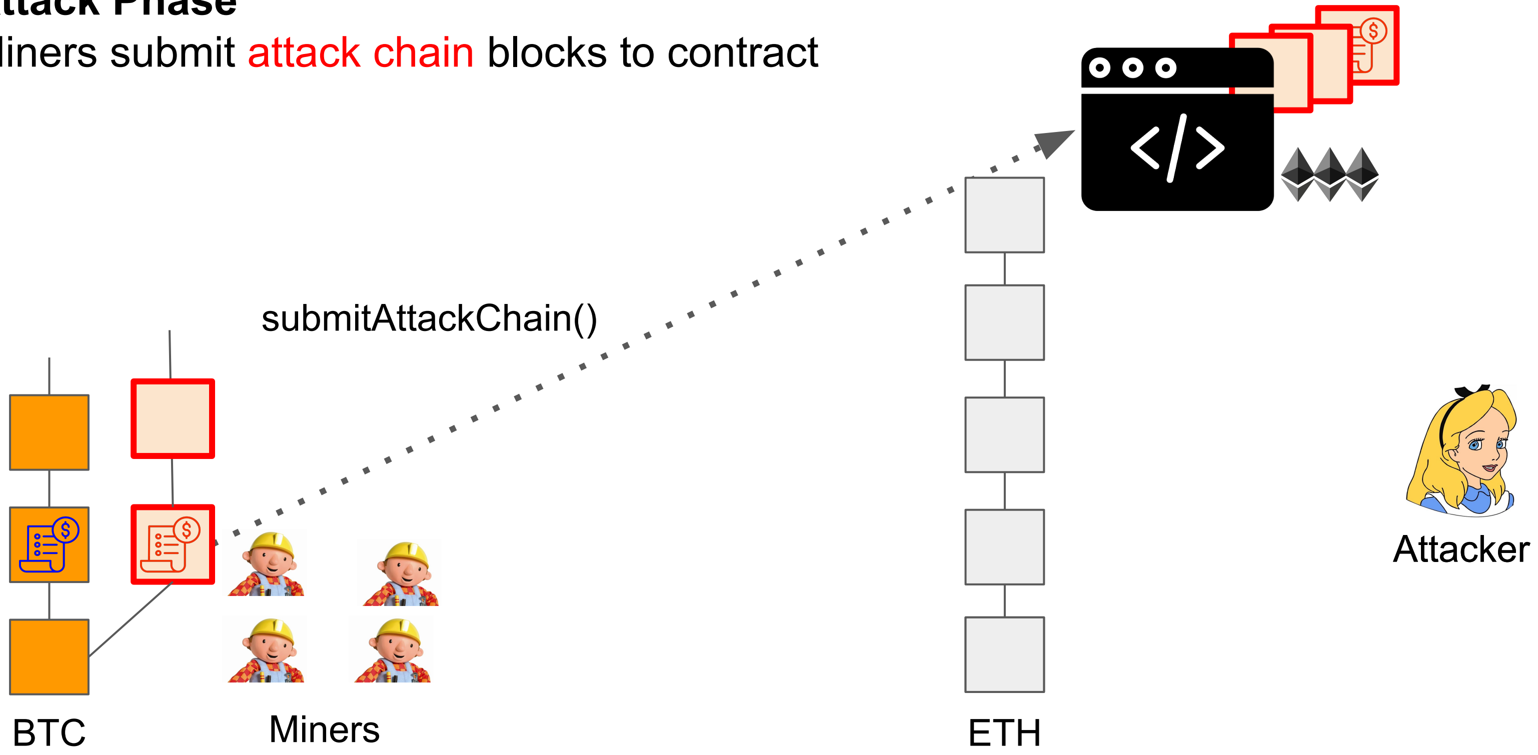
# Pay-to-Win Attack (Deep Fork)

**Attack Phase**:
- Miners mine on block templates, executing the attack
- Attacker can extend the attack (new templates + funds)



# Pay-to-Win Attack (Deep Fork)

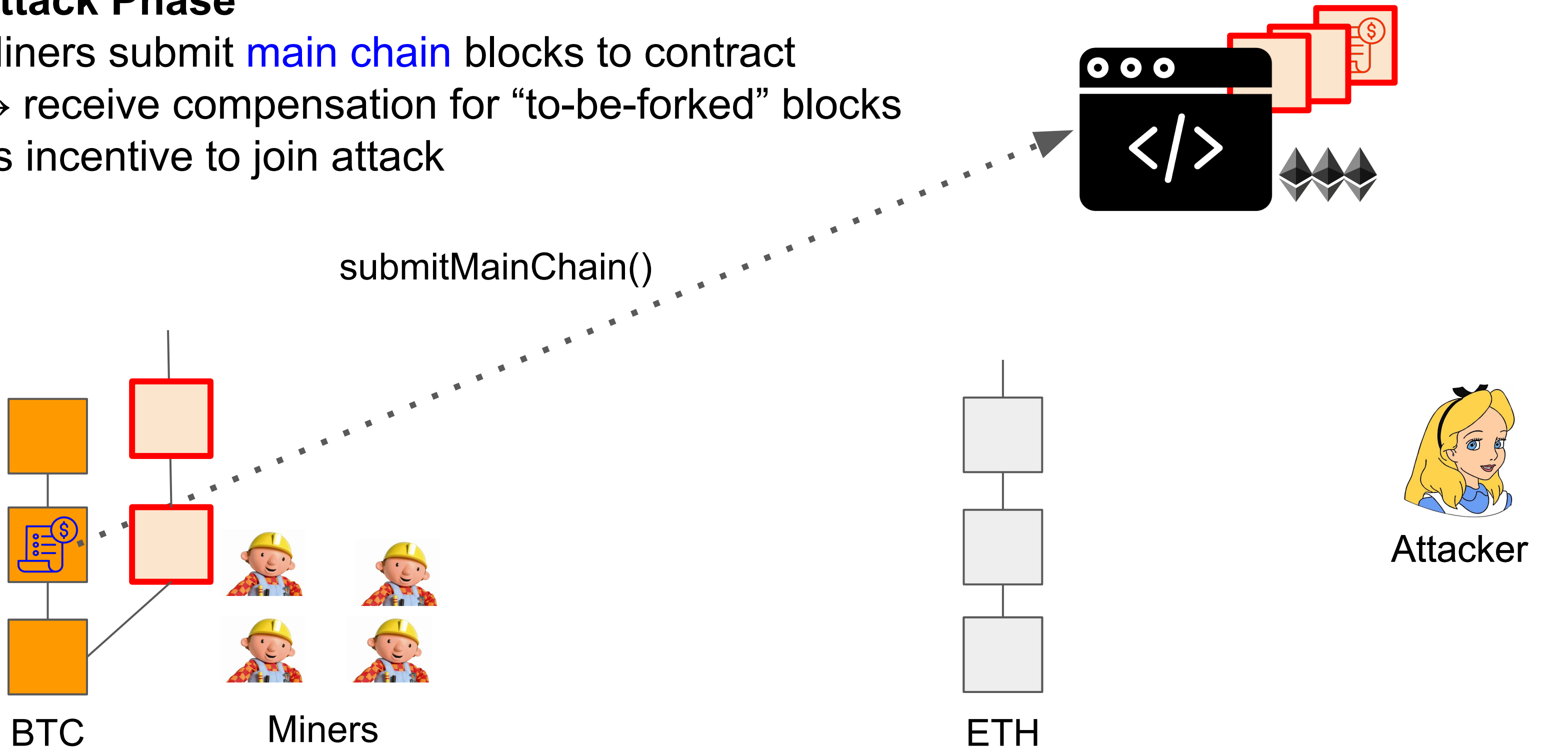
**Attack Phase**
Miners submit attack chain blocks to contract



# Pay-to-Win Attack (Deep Fork)

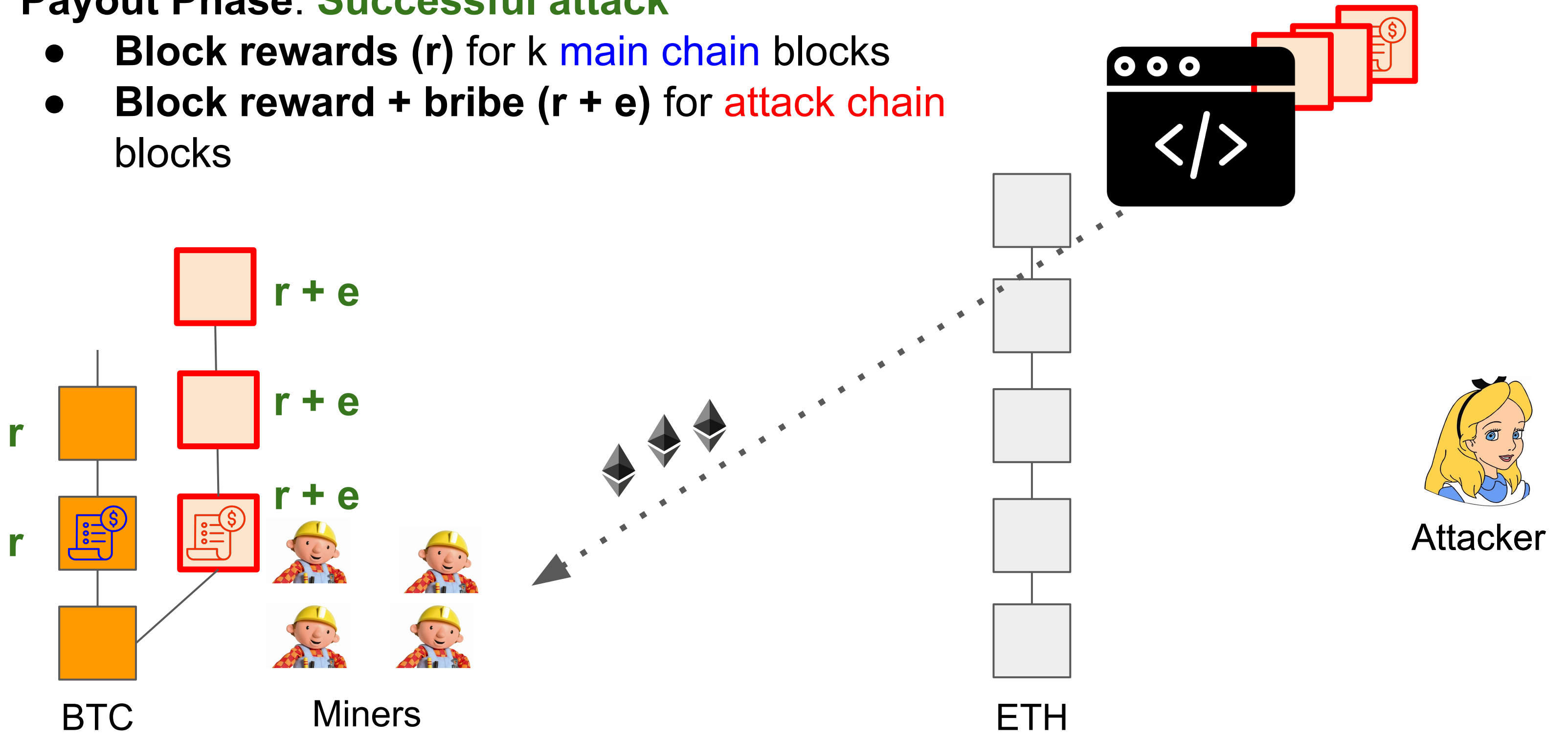
**Attack Phase**

Miners submit main chain blocks to contract
→ receive compensation for “to-be-forked” blocks
as incentive to join attack



# Pay-to-Win Attack (Deep Fork)

**Payout Phase**: **Successful attack**
- **Block rewards (r)** for k main chain blocks
- **Block reward + bribe (r + e)** for attack chain blocks

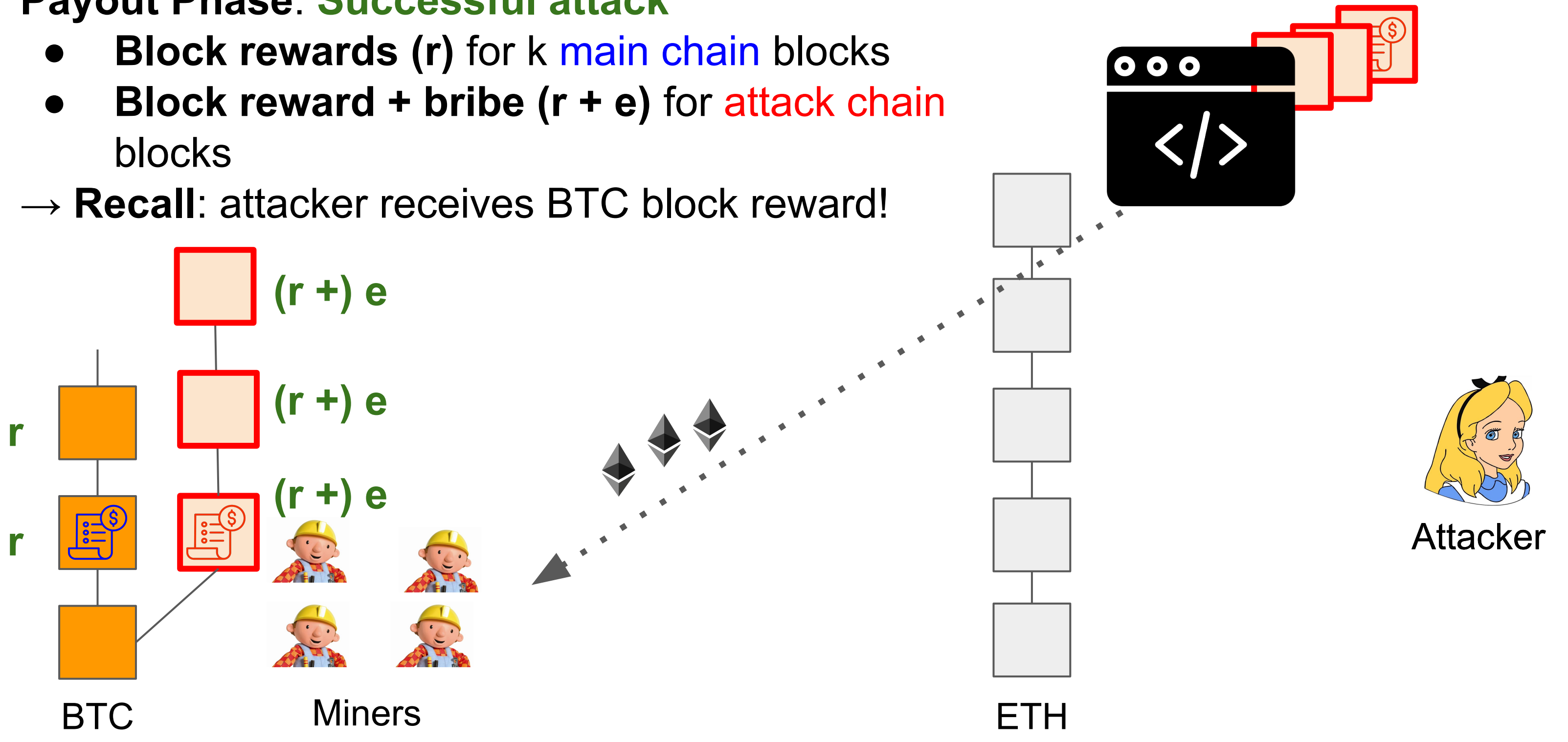


# Pay-to-Win Attack (Deep Fork)

**Payout Phase**: **Successful attack**

- **Block rewards (r)** for k main chain blocks
- **Block reward + bribe (r + e)** for attack chain blocks

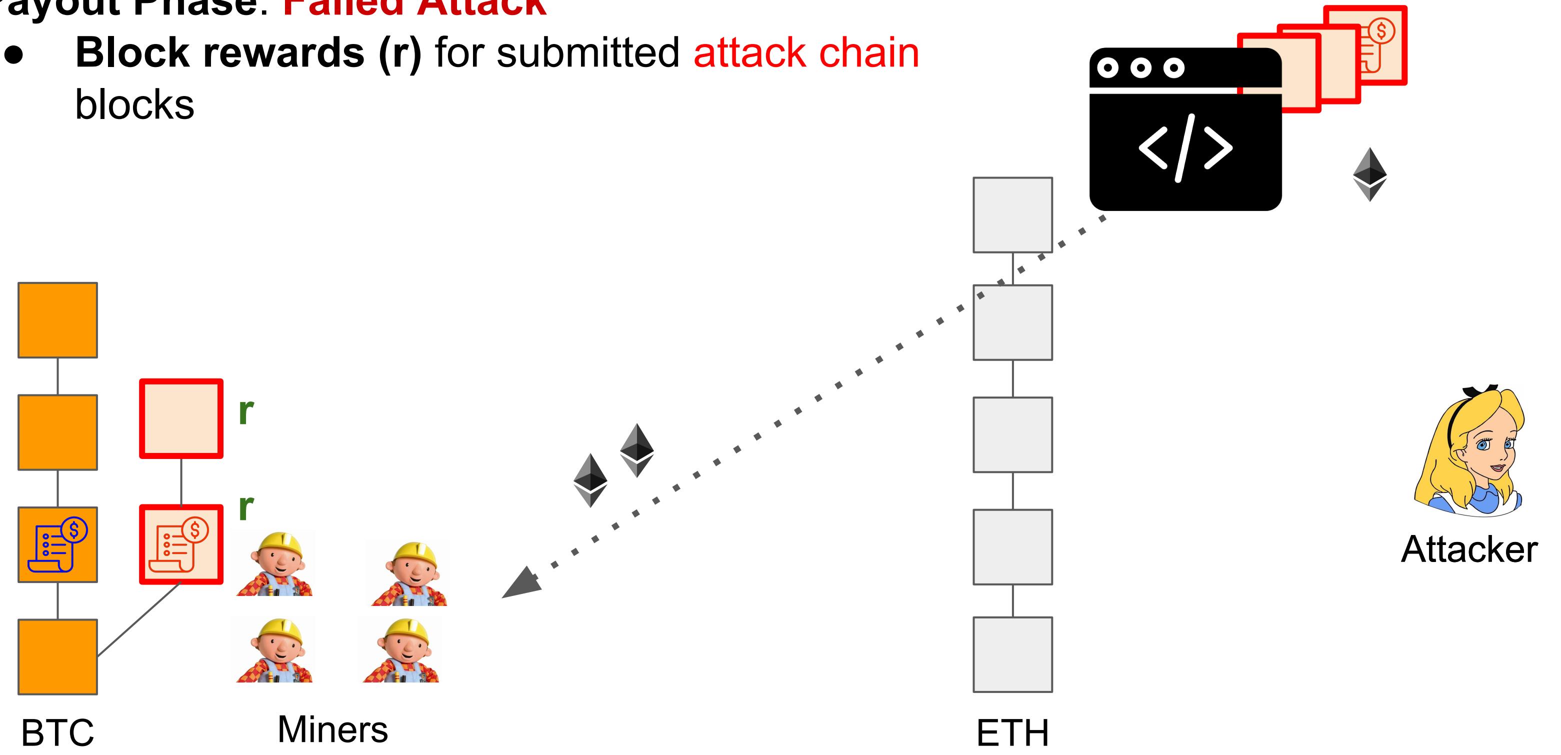
→ **Recall**: attacker receives BTC block reward!



# Pay-to-Win Attack (Deep Fork)

**Payout Phase**: **Failed Attack**

- **Block rewards (r)** for submitted attack chain blocks



# Pay-to-Win Attack (Deep Fork)

**Required funds at the start of attack:**

$$N * (e + r) + k * r$$

**N** … attack duration
**e** … bribe
**r** … block reward
**k** … confirmation required by victim

# Cost Evaluation

$k = 6$ (min. 6 main chain + 7 attack chain blocks to succeed )

$r = 14$ *BTC* *(~ block reward)*

$e = 1$ *BTC* (bribe - can be set way lower!)

**Rational miners only (no victim hash rate)**

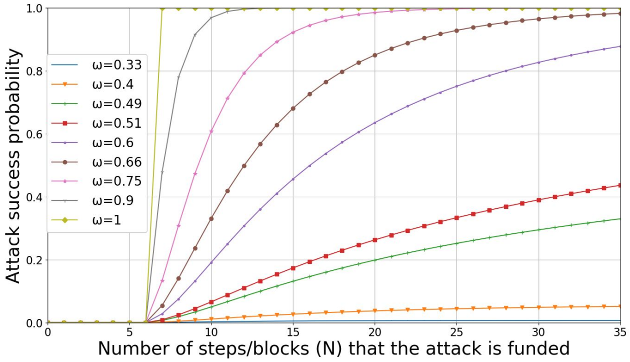
- Failed attack ~ 98 BTC
- Successful attack ~ 91 BTC

# Cost Evaluation

*k = 6* (main chain must have 6 blocks before double spend succeeds)
*r = 14 BTC* *(~ block reward)*
*e = 1 BTC* (bribe - can be set way lower!)

**Altruistic miners (victim has hash rate)**

| $\omega$ | whale costs | p2w costs $c_{failed}$ (worst case lose) | % whale | p2w costs $c_{success}$ (worst case win) | % whale | p2w costs (expected win) |
|---|---|---|---|---|---|---|
| 0.532 | 2.93e+23 | 7305 | 0.00 | 577 | 0.00 | 144 |
| 0.670 | 999.79 | 600 | 60.01 | 130 | 13.00 | 104 |
| 0.764 | 768.09 | 330 | 42.96 | 112 | 14.58 | 100 |
| 0.828 | 1265.14 | 240 | 18.97 | 106 | 8.38 | 99 |
| 0.887 | 1205.00 | 195 | 16.18 | 103 | 8.55 | 98 |
| 0.931 | 1806.67 | 165 | 9.13 | 101 | 5.59 | 98 |
| 0.968 | 2178.58 | 135 | 6.20 | 99 | 4.54 | 97 |
| 0.999 | 2598.64 | 120 | 4.62 | 98 | 3.77 | 97 |




See paper for more details!

# Pros and Cons

+ **Difficult to detect** (cross-chain)
  → monitor all smart contract chains?
+ Miners have **no risk**
+ Only **small bribes** necessary
+ **No trust** required between attacker and miners

- Requires **smart contracts** on funding chain
- Funding chain must be able to **verify PoW** of target chain
- **Exchange rate** handling

# Crowdfunding

- Use smart contract to coordinate multiple attacks in parallel
- Attackers lock in
  - e.g. double spend TX
  - compensation
- Attack costs are typically fixed!
  - Split among participants

**Challenges**: timing, sabotage via conflicting attacks, ...

See paper for more discussion!

# Implications: Transaction Security

Typically, we assume a global ***k*** (Backbone model)

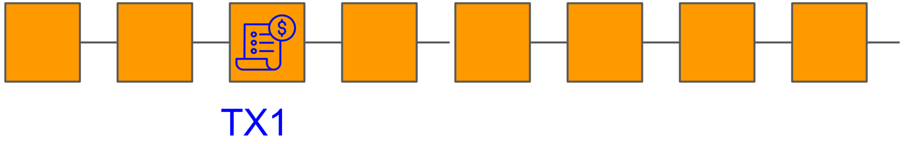
Sompolinksy et al. argue: “Take into account TX value!”

**Recently:**

Zindros argues: “Take into account value of entire block!”

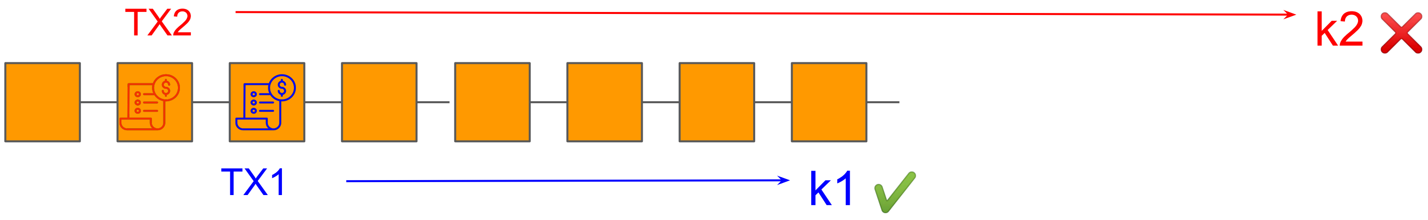
**We conjecture:** Even this is insufficient!

# Implications: Transaction Security




Value of block of TX1 → set k1 (e.g. 6)

# Implications: Transaction Security




Value of block of TX1 → set k1 (e.g. 6)

**Problem**: “juicy” TX2 in prev. block with high value being attacker

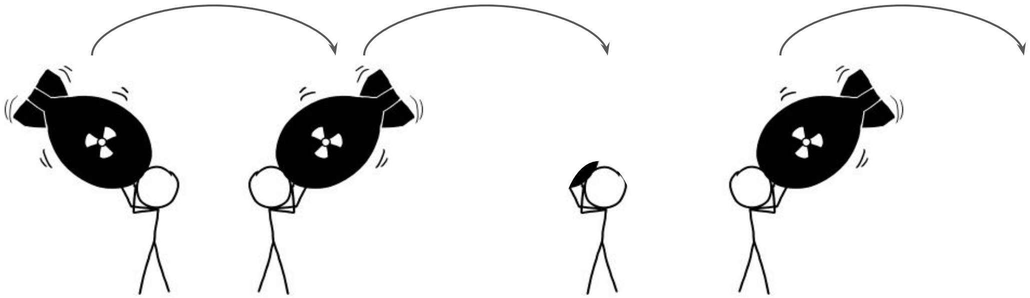
- k1 sufficient for TX1 alone… but what if the attack on TX2 **occurs anyway**?
- What if attacker of TX2 could also attack TX1 as “extra”?

→ In practice: **crowdfunded attacks**

# What To Do? (Take With a Grain of Salt)

From theoretical perspective:

**“HODLING” is risky!**



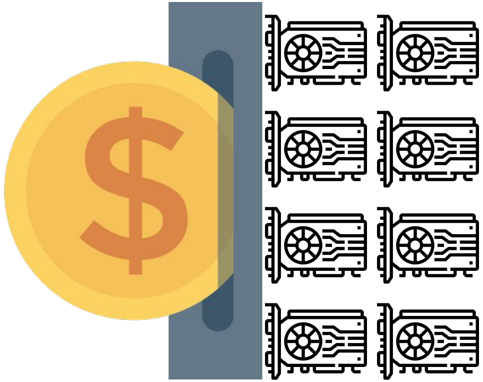
**Only “safety” measure**:

As soon as you receive coins → spend & transfer risk!

This is theory! Less of a problem in practice.

Devcon 5 Research Meetup

11 October 2019, Osaka, Japan



# Questions?

**Pay-to-Win: Incentive Attacks on Proof-of-Work Cryptocurrencies**

*Aljosha Judmayer, Nicholas Stifter, Alexei Zamyatin, Itay Tsabary, Ittay Eyal, Peter Gazi, Sarah Meiklejohn, and Edgar Weippl*