Speculative Denial-of-Service Attacks in Ethereum

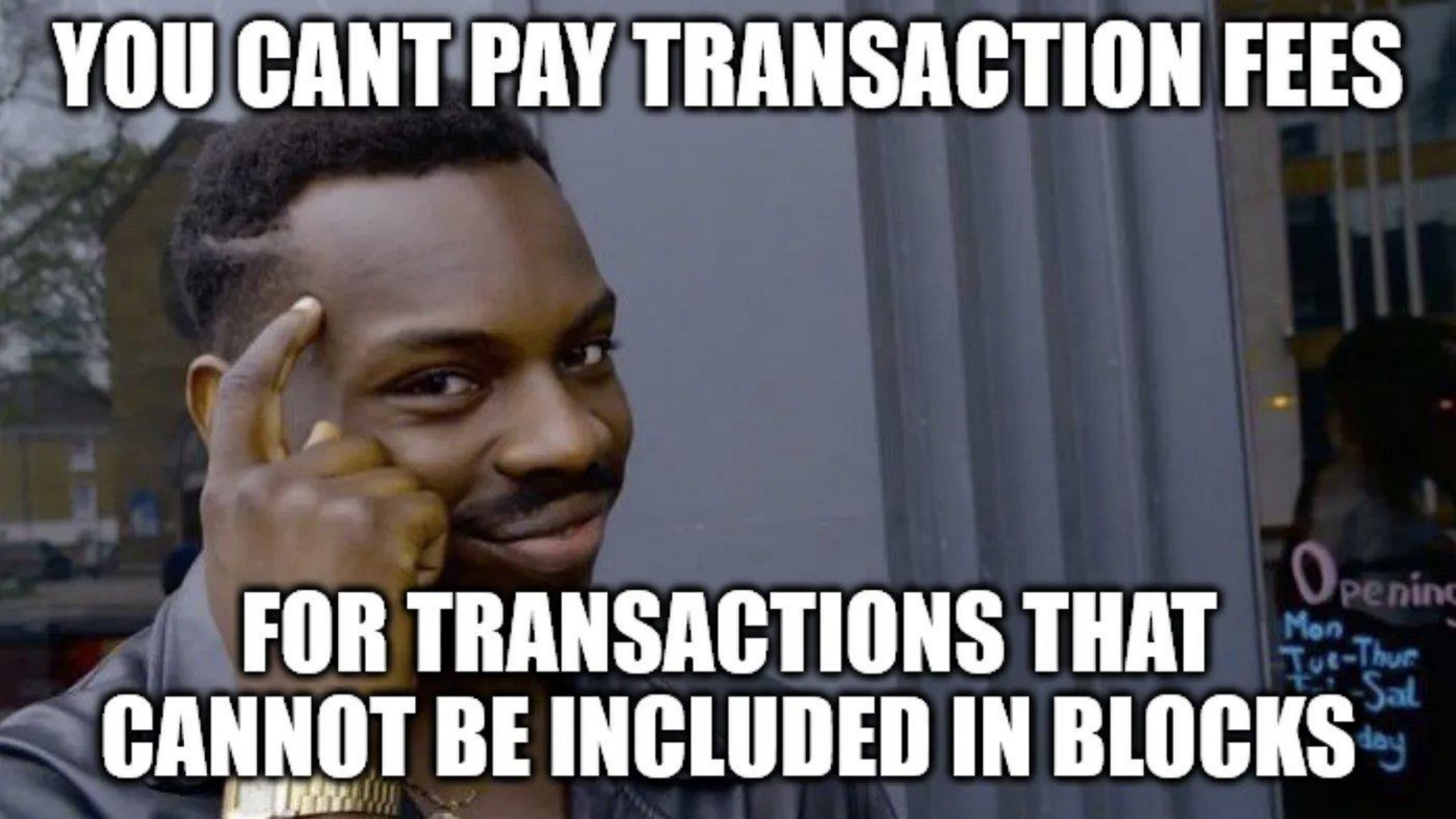
Aviv Yaish, Kaihua Qin, Liyi Zhou, Aviv Zohar, Arthur Gervais





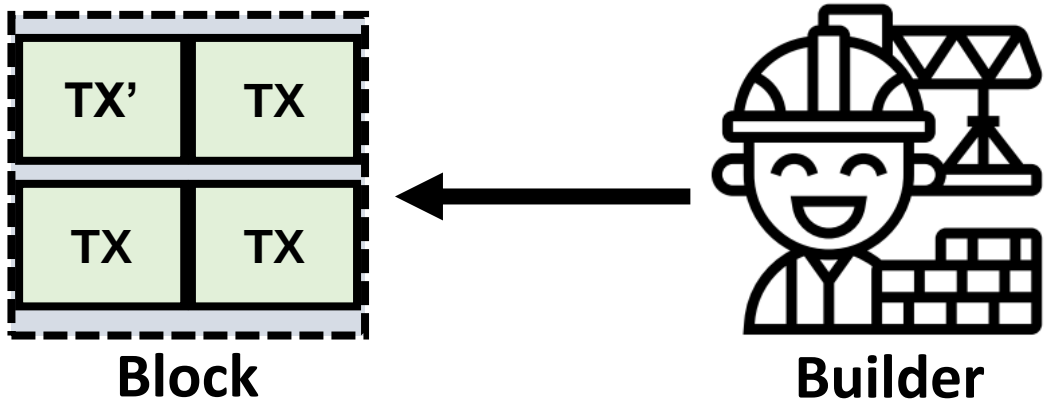
Background: **Transaction Fees**

- We attack Ethereum's transaction fee mechanism
- Ethereum transactions may execute arbitrary code
  - Each unit of computation is measured in gas
  - TXs that enter the blockchain pay fees per gas unit consumed
- TXs can *revert*: roll-back any actions they've made
  - Even reverted transactions pay fees, to prevent DoS attacks
  - **Is this enough?**


YOU CANT PAY TRANSACTION FEES
FOR TRANSACTIONS THAT CANNOT BE INCLUDED IN BLOCKS

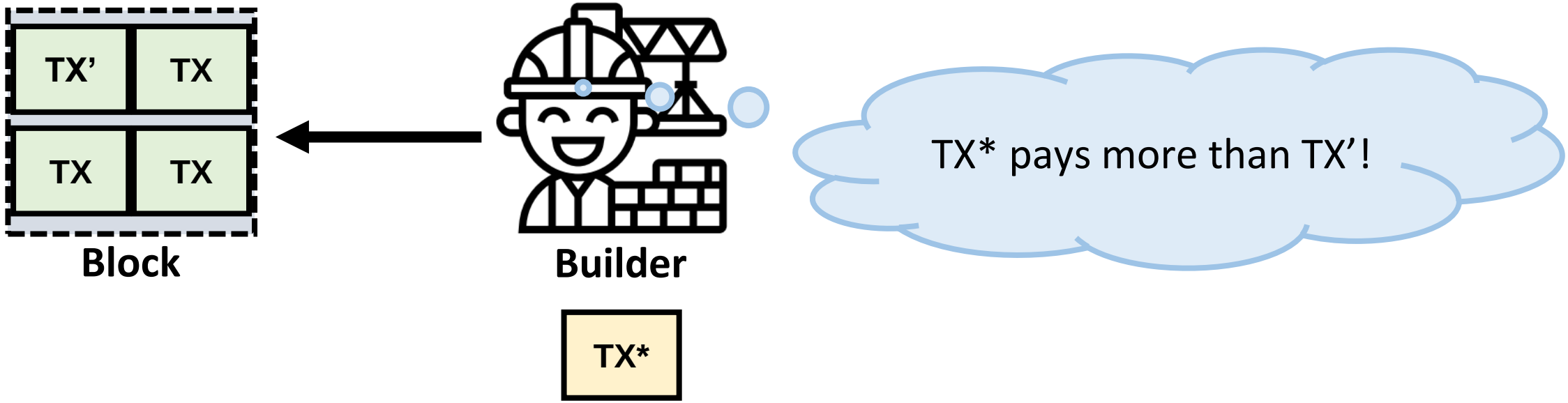
Insight: Speculative Resource Investment

- Actors *speculatively* invest computational resources in TXs
- E.g., block builders may execute more TXs than can fit in one block



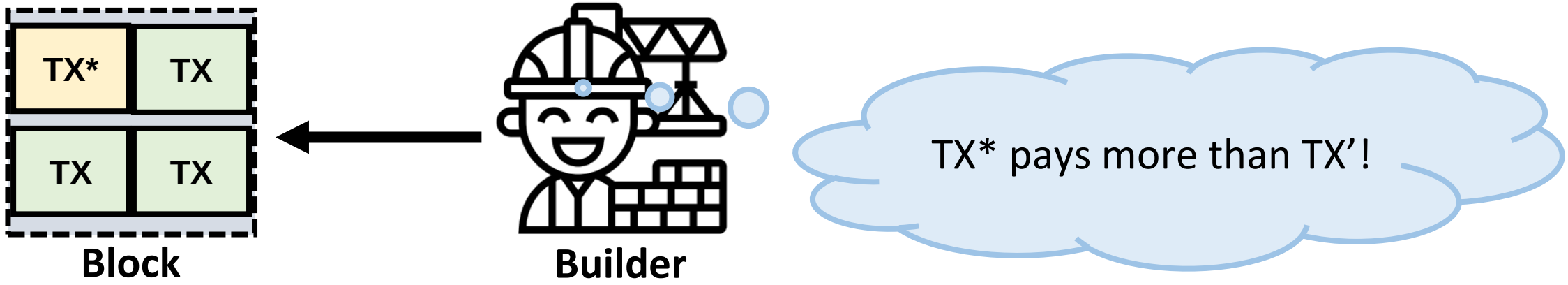
Insight: Speculative Resource Investment

- Actors *speculatively* invest computational resources in TXs
- E.g., block builders may execute more TXs than can fit in one block



Insight: Speculative Resource Investment

- Actors *speculatively* invest computational resources in TXs
- E.g., block builders may execute more TXs than can fit in one block

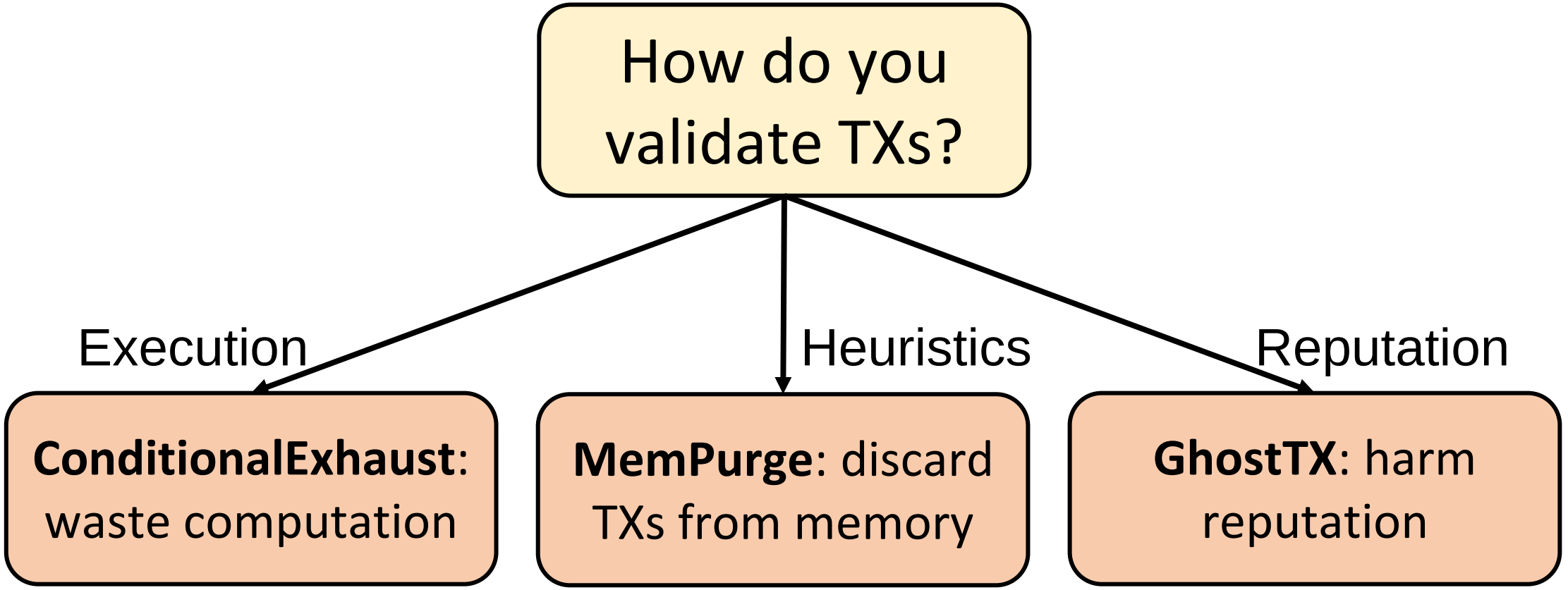


No payment for TX'

Overview

- **Goal:** lower the revenue of competing blockchain actors
- **Method:** we present multiple attacks that
  - Waste victim resources on *invalid* TXs that appear *valid*
  - Circumvent defenses that ensure TX validity (hard!)
  - Tested on geth, the most popular Ethereum execution client
  - Applicable to other cryptocurrencies
- Mitigations necessitate trading off security with user experience
  - Blockchain security relies on ***much more*** than consensus mechanisms

Our **Attacks**



Summary: you're damned if you and damned if you don't!

Attack 1: **ConditionalExhaust**

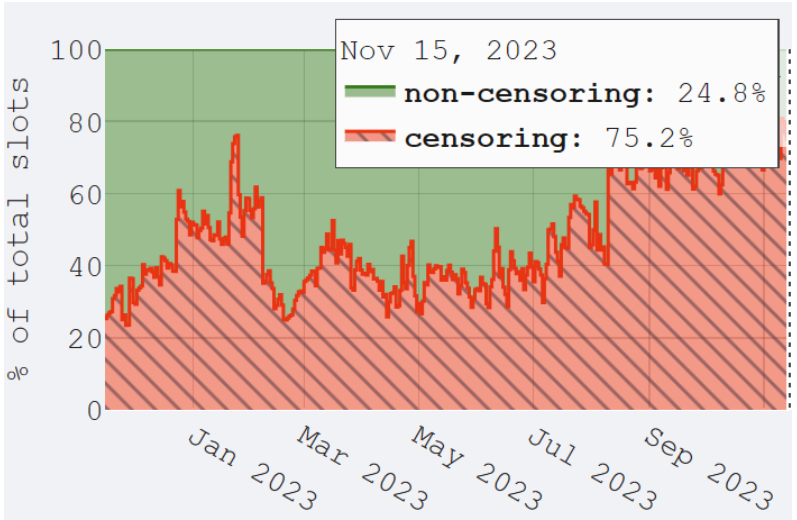
- When creating blocks, actors must execute TXs
- Trick victims to waste time on TXs that cannot be included in blocks
  - Create *invalid* TXs that appear *lucrative*
  - → attack TXs are processed before other TXs
  - → attack TXs cannot be included in blocks
  - → victims' revenue is harmed

OFAC **Sanctions**

- Sanction compliant actors censor non-compliant TXs
- Censorship is *“local”*: not enforced by consensus
- Compliant actors cannot collect fees from non-compliant TXs

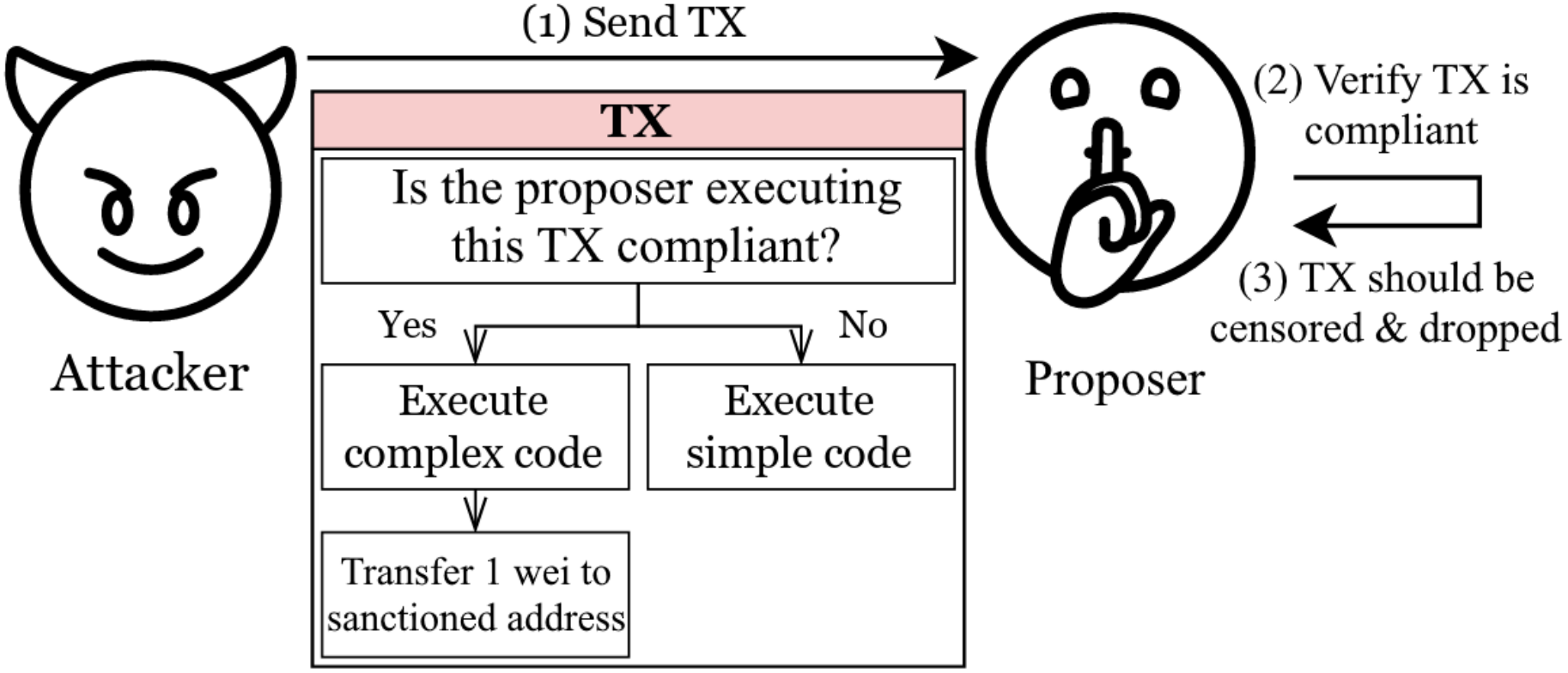
Source: https://home.treasury.gov/news/press-releases/jy0916

Source: https://censorship.pics/



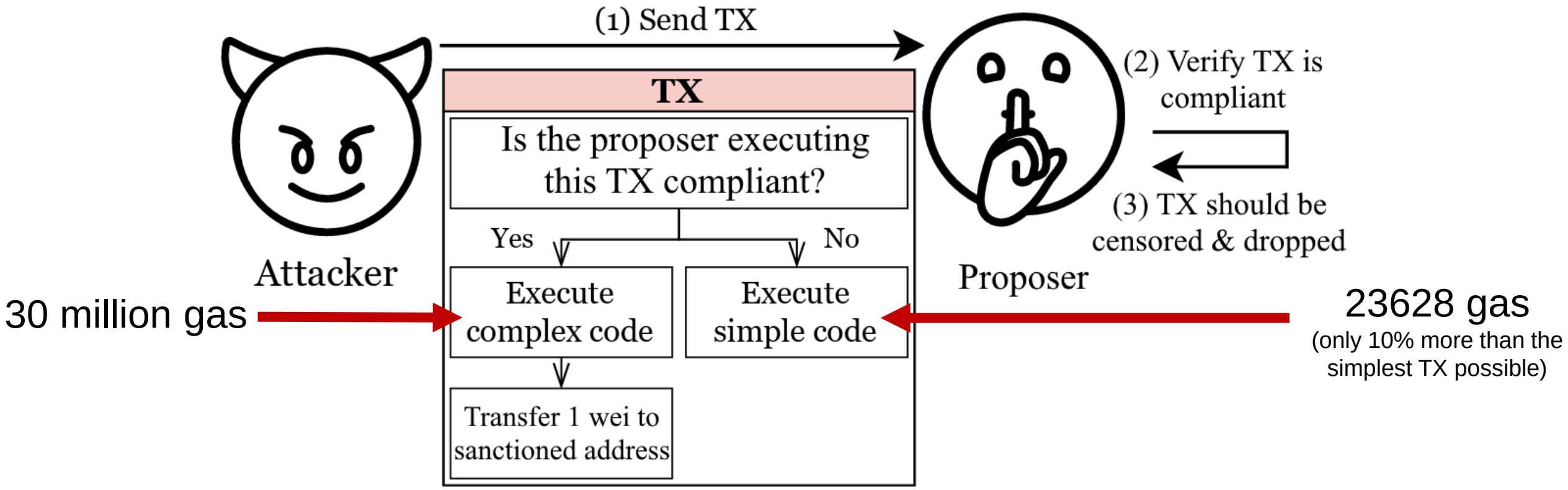
ConditionalExhaust: **Censorship** Variant

- Send attack TXs when proposer is compliant & cannot include them
- We present variants that do not rely on censorship (see paper)



ConditionalExhaust: **Censorship** Variant

- If an attack TX is included, it pays for ~1270x less gas than used



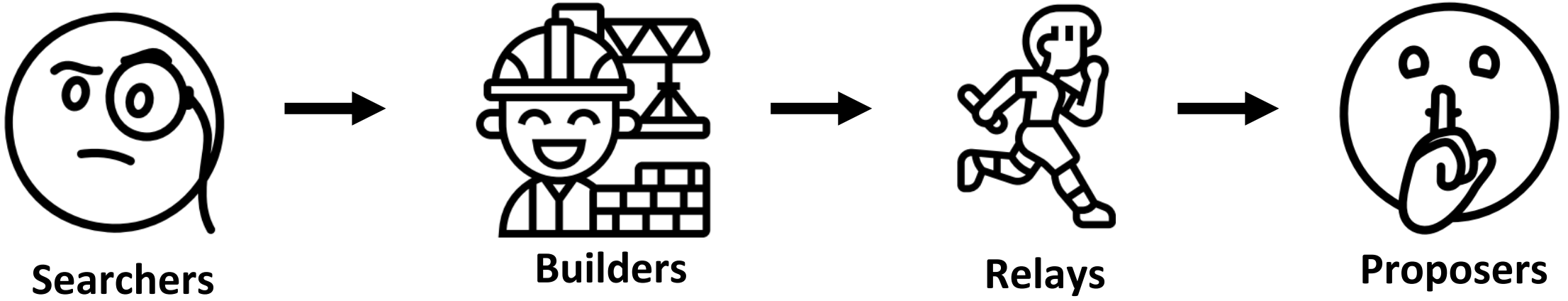
Evaluation

- Flashbot's min specs: 4 core CPU @ 2.8GHz, 16GB RAM, SSD
- Our testbed's specs: 64 core CPU @ 2.9GHz, 256GB RAM, NVMe SSDs
- 140 TXs cause testbed to mine empty blocks
- Total cost: at most $770 for TXs to be prioritized over 90% of TXs

Attack 2: **MemPurge**

- MemPurge tricks victims to store invalid TXs in their memory
  - Creates *invalid* TXs that heuristics find *valid* (without censorship)
  - → other TXs are evicted to make room for attack TXs
  - → attack TXs cannot be included in blocks
  - → victims' revenue is harmed
- This is ***hard***! Geth has a thick layer of defenses
- We circumvent them via a multi-phased attack (see paper for details)

Background: **Proposer Builder Separation (PBS)**



Searchers

Collect TXs in bundles & extract value from them

Builders

Pack TXs & bundles in blocks

Relays

Verify block contents

Proposers

Propose blocks

Attack 3: **GhostTX**

- DoS risk: creators of invalid blocks/bundles do not pay fees
- Some builders prioritize searchers with good “reputation”
  - Meaning, searchers whose TXs tend to enter the chain
- GhostTX tricks searcher victims to include attack TXs in bundles
  - → attack TXs are invalidated by the attacker
  - → victim reputation is decreased
- First attack on the PBS ecosystem (see paper for details)

Conclusion: **Call to Arms**

- TX validation: free-for-all, ripe for future work
  - Other proposed mechanisms rely on speculation
  - Are they vulnerable?
- We present more attack vectors, read our paper!
  - E.g.: future proposer duties are known in advance
  - Prior work: future proposers can be attacked
  - This work: future proposers ***can attack***
  - Can these attacks be prevented?

Thank you!

Code

https://github.com/AvivYaish/SpeculativeDoS

Paper

https://ia.cr/2023/956

Reach out: aviv@avivyaish.com