Background

Smart contracts are crucial elements of decentralized technologies, but they face significant obstacles to trustworthiness due to security bugs and trapdoors. To address the core issue, we propose a technology that enables programmers to focus on design-level properties rather than specific low-level attack patterns. Our proposed technology, called Theorem-Carrying-Transaction (TCT), combines the benefits of runtime checking and symbolic proof. Under the TCT protocol, every transaction must carry a theorem that proves its adherence to the safety properties in the invoked contracts, and the blockchain checks the proof before executing the transaction. The unique design of TCT ensures that the theorems are provable and checkable in an efficient manner. We believe that TCT holds a great promise for enabling provably secure smart contracts in the future.

Motivation

This proposal is necessary since the Ethereum protocol does not ensure the safety features on the design level. It stems from the recognition of the significant obstacles faced by smart contracts in terms of trustworthiness due to security bugs and trapdoors. While smart contracts are crucial elements of decentralized technologies, their vulnerabilities pose a challenge to their widespread adoption. Conventional smart contract verification and auditing helps a lot, but it only tries to find as many vulnerabilities as possible in the development and testing phases. However, in real cases, we suffer from the unintentional vulnerabilities and logical trapdoors which lead to lack of transparency and trustworthiness of smart contract.

Reference

Technical WhitePaper: https://arxiv.org/ftp/arxiv/papers/2304/2304.08655.pdf

Demo Repo: GitHub - TCT-web3/demo: The first demo of TCT