DISTRIBUTED VERIFICATION OF SMART CONTRACTS

AUTHORS: Chad E. Brown, Ondřej Kunčar, Josef Urban

BLOCKCHAIN

Cryptocurrencies—steady increase
Based on two technologies:

- Blockchain
- Distributed consensus protocol

Main appeal: no trusted central authority

Bitcoin: send money from A to B
Ethereum: smart contracts (Turing-complete language)

TheDAO Hack: June 2016, $60M lost

DISTRIBUTED PROOF MARKET

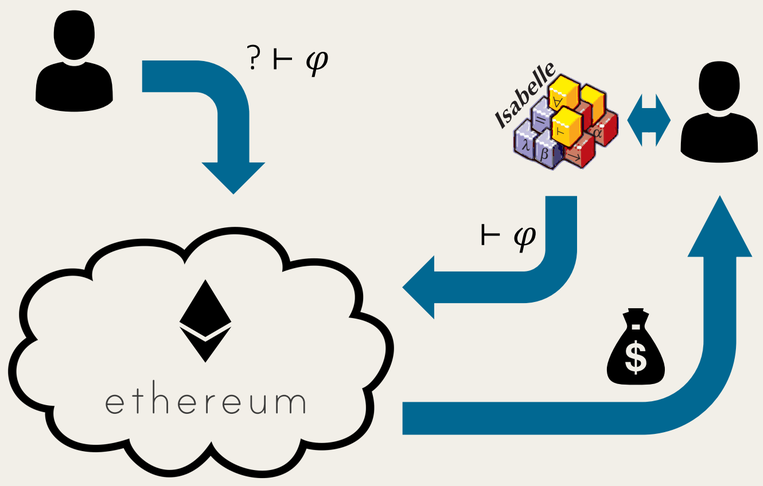
Writing proofs is difficult
Flyspeck:

- 500k lines of proofs
- 20 person-years

A lot of energy needed for:

- Finding manpower
- Organizing manpower

Solution: proof market
Makes sense only if decentralized
➔ smart contracts



CONTRACTS VERIFICATION

Contract in Isabelle/HOL
(shallow embedding)

Prove high-level properties
(“clients can’t loose money”)

Refine to a low-level specification
(Refinement Framework)

Prove low-level properties
(“the stack can’t overflow”)

Ethereum bytecode
(deep embedding)

The contract = simple proof checker
+ commitment scheme

Metamath’s proof checker:

- 300 lines in Python
- 20k top level lemmas = 100M string operations
- Ethereum (2015) = $20k
- Ethereum (2017) = $2M

Previous work: ProofMarket, Mathgate, Qeditas

DISTRIBUTED VERIFICATION

Useful when we combine distributed proof market and smart contracts verification

Synergy between these two technologies can foster their further development

CZECH INSTITUTE OF INFORMATICS ROBOTICS AND CYBERNETICS CTU IN PRAGUE

Technical University of Munich