Ethereum: State of Knowledge and Research Perspectives

Sergei Tikhomirov

SnT, University of Luxembourg

24 October 2017
LORIA, Nancy, France

Outline

Motivation

Technical overview

Open problems

- Cryptography
- Consensus
- Scalability
- Privacy
- Contract programming
- Other issues

Conclusion

Why give this talk?

- Ethereum is a fascinating research topic
- Intersection of cryptography, distributed systems, programming languages, privacy, game theory, ...
- Interesting problems of highest practical relevance

Bitcoin

- A fully decentralized digital currency [Nakamoto 2008]
- Combines cryptography and economics to prevent double spending without a trusted third party

Ethereum: generalized blockchain

- A blockchain-based application platform [Buterin 2014]

- Key feature: Turing complete programming

Ethereum features

- Accounts controlled by key or by code (*smart contracts*)
- Developers write contracts in high-level languages that compile to Ethereum Virtual Machine (EVM) bytecode
- Users interact with contracts via transactions (e.g., send *ether*, perform computation)

Ethereum security is hard

- New software stack
- Unfamiliar execution paradigm
- Very limited ability to patch contracts
- Anonymous financially motivated attackers
- Rapid pace of development

Outline

Motivation

Technical overview

Open problems

- Cryptography
- Consensus
- Scalability
- Privacy
- Contract programming
- Other issues

Conclusion

Cryptography

- Signatures: ECDSA
- Hash for id's: Keccak-256
- Hash for proof-of-work: **Ethash**

Ethash

- A new memory-hard cryptographic hash function
- Developed in 2013–2015, no academic cryptanalysis
- Claims of weaknesses in early versions
- (Ethereum plans to abandon proof-of-work altogether)

Outline

Motivation

Technical overview

Open problems

- Cryptography
- Consensus
- Scalability
- Privacy
- Contract programming
- Other issues

Conclusion

Consensus: proof-of-work

- Nodes (miners) compete to produce the next block
- Find *nonce* s. t. $hash(nonce|blockheader) < target$
- The first miner to construct a block gets a reward
- Probability of success is proportional to hashing power

Drawbacks of proof-of-work

- Energy consumption (Bitcoin: #70 ”country”)
- Centralization (benefits from economies of scale)
- Game-theoretic attacks (selfish mining)

These problems are less obvious in Ethereum than in Bitcoin.

Proof-of-stake as "virtual mining"

Validators chosen proportionally to *stake*. Known issues:

- Nothing-at-stake (incentive to mine on all chains)
- Choosing validators (security of randomness source)
- Long-range attacks (finality guarantees)

Independent evaluation required!

Outline

Motivation

Technical overview

Open problems

- Cryptography
- Consensus
- Scalability
- Privacy
- Contract programming
- Other issues

Conclusion

Scalability

- Ethereum: 10 tx/sec (Visa: 45k tx/sec)
- Proposed solution: payment channels
- Exchange partially signed tx's off-chain, settle on-chain
- Payment channel network Raiden is in development
- Related: sharding, fast synchronization

Outline

Motivation

Technical overview

Open problems

- Cryptography
- Consensus
- Scalability
- Privacy
- Contract programming
- Other issues

Conclusion

Privacy

- All transactions in plaintext, history stored forever
- Blockchain analysis, deanonymization (mostly Bitcoin)
- Possible solution: ZKP / ZkSNARKs (used in ZCash)
- Introduced in Ethereum on 16 October 2017

Outline

Motivation

Technical overview

Open problems

- Cryptography
- Consensus
- Scalability
- Privacy
- Contract programming
- Other issues

Conclusion

Contract programming in Solidity

Solidity is the most mature high-level contract language. Example of a simple program:

```
pragma solidity 0.4.17;
contract StringStorageContract {
  string private str = "Hello, world!";
  function getString() public constant
  returns (string) {
    return str;
  }
  function setString(string _str) public {
    str = _str;
  }
}
```

Improving code quality

- Summarizing good and bad practices
- Developer tools: code analysis, bug detection
- Formal verification, formalization of EVM
- Safer paradigms, languages, frameworks

Outline

Motivation

Technical overview

Open problems

- Cryptography
- Consensus
- Scalability
- Privacy
- Contract programming
- Other issues

Conclusion

Other issues

- Governance: who determines Ethereum's future?
- Usability: friendly dApps for broader audience
- Ethical: what is responsible disclosure in blockchain?
- Legal: how do cryptocurrencies fit in legal systems?

Conclusion

- Blockchain is still a new technology
- Ethereum poses many research challenges
- Potential is enormous
- Security issues are inevitable

Researchers are welcome!

Questions?

- cryptolux.org
- s-tikhomirov.github.io

UNIVERSITÉ DU
LUXEMBOURG