Ethereum Analysis: Gas Economics and Proof of Work

Overview

From January through March of 2015, Least Authority, conducted an analysis of Ethereum's Gas Economics and Proof of Work, in a consulting capacity with DEVolution, GmbH. The two analysis topics are independent and presented here as separate documents comprising this report:

- GasEcon.md : An analysis of the Ethereum Virtual Machine (EVM), its use of "gas" to compensate miners for the computational costs of verification, and the resulting economic effects. Includes an examination of contract composition and defensive programming strategies.
- PoW.md : An analysis of "Ethash", the Proof-Of-Work function which makes it difficult and expensive to attack Ethereum's consensus algorithm. We examine its goals ("GPU-friendly, ASIC-resistant"), compare it to existing techniques in the password-hashing literature, and discuss how it will need to evolve as conditions change.
- Appendix.md : a summary of mitigations applied since delivering our report

Key Results

Gas Economics

- Ethereum's "gas" system *is* likely to meet its primary goals: enabling complex transactions while mitigating certain Denial-of-Service attacks.
- Changes will be needed in the future, when the system grows and gas fees replace block rewards as the primary income for miners.
- Contracts *can* be composed, but safe cooperation between mutually-distrusting parties will require careful study and rigorous defensive programming. Some changes to the virtual machine could be made to improve the safety of these compositions, and higher-level analysis tools must be developed. The

programming examples included in serpent and found in the wild are flawed and inadequate for demonstrating best practices.
- The rule that "value remains with the recipient upon exception" is particularly surprising and difficult to work around. We highly recommend changing this to "value reverts to the sender upon exception".

Ethash Proof-of-Work

- Ethereum introduces a novel *Ethash* Proof-of-Work algorithm.
- Its high level design goals, summarized as "GPU-friendly, ASIC-resistant" are reasonable towards achieving its intended economic effect.
- Ethash is based on random paths over a fixed read-only dataset, which is closely related to prior work in academic literature on memory-bound Proof-of-Work puzzles and one Password Hashing Competition finalist.
- The custom randomizing functions used are non-standard and difficult to cryptanalyze, but they hold up to simple statistical tests.
- The concrete parameterization is also reasonable based on our analysis of currently available hardware.
- There is a social contract to transition away from Proof-of-Work altogether, further reinforcing a goal of making specialized hardware development economically infeasible.
- We are concerned about the scheduled dataset growth. We believe this mechanism is ineffective at achieving its intended effect, but not terribly harmful.

Background

Ethereum

Ethereum is a decentralized consensus network for processing smart contracts. Ethereum is typically considered a descendent of Bitcoin, a predecessor which introduced decentralized consensus for transaction processing using a *Proof-of-Work* mechanism, with scriptable transactions to specify a wide range of contractual agreements.

The fundamental distinction between Bitcoin and Ethereum is that while the former provides complex transaction criteria which can codify various contractual rules, the latter allows for /stateful/ code instances called *contracts* which persist across multiple transactions and whose code can be Turing complete and use custom state transitions

to codify contractual agreements. Ethereum introduces a *gas* abstraction as a system of accounting for storage and processing of these contracts by the network verifiers.

Although stateful contracts using gas is the fundamental design space difference Ethereum has from Bitcoin, it has *many* other practical differences in design, including a distinct Proof-of-Work consensus system. This alternative consensus system relies on a new custom Proof-of-Work algorithm designed by the Ethereum project called *Ethash*. Its design goals attempt to limit the marginal advantage of developing specialized hardware for consensus *mining*, while still allowing lightweight clients to verify the "weight" of candidate chains.

This report focuses on the gas accounting system and its economic impact on the system, as well as the custom Ethash algorithm.

DEVolution, GmbH

DEVolution, GmbH is Ethereum's Swiss development organization. They have commissioned this report in order to evaluate and improve the security of Ethereum in preparation for its anticipated release during 2015.

Least Authority

Least Authority, LLC, is a private company specializing in privacy preserving, decentralized, open source, and user-empowering technologies. We've performed security audits and provided security and cryptography engineering consultation for a variety of open source privacy or transparency related projects, including CryptoCat, GlobalLeaks, SpiderOak, and Ooni

Additionally, Least Authority employs most of the core developers of the Tahoe-LAFS decentralized secure data storage network.

This review was conducted by Andrew Miller, Brian Warner, and Nathan Wilcox.

Scope

Method

In order to be maximally useful during Ethereum's development, we have made our findings available to Ethereum developers informally and incrementally as quickly as possible.

A github repository containing our final report will be made publicly available at https://github.com/LeastAuthority/ethereum-analyses and incudes additional code examples and tools.

Coverage and Sources

As our report comprises a *design analysis* rather than an *implementation* security audit, our primary sources are the written specifications, especially the "Yellow paper" specification. However, we also made use of source code in order to improve our understanding and develop illustrative examples. Although we took snapshots of the specifications and code at the beginning of our analysis, we also attempted to use the most recent versions as updates occurred. When we have needed to reference specific prior versions, we have indicated the version or commit hash inline in our report.

Our primary sources:

- the yellow paper
- the Go-language implementation and pyethereum
- serpent programming examples
- the ethash wiki article

document source

This file generated from https://github.com/LeastAuthority/ethereum-analyses