Master Thesis Proposal

Identifying Semantic Properties in the Bytecode of Smart Contracts Using Symbolic Execution

David Loz, BSc

12123562

Faculty of Informatics
Vienna University of Technology

Advisor: Gernot Salzer

March 10, 2023

1 Motivation and Problem Statement

The blockchain technology has received increased attention in recent years, with Ethereum being one of the most prominent projects in this area. It is a blockchain with a fully fledged Turing-complete programming language that enables the creation of blockchain programs (also known as "smart contracts") [1]. Triggered by external transactions, these contracts are executed on the Ethereum Virtual Machine (EVM) and can interact with other contracts or accounts, allowing for the creation of decentralized applications on the Ethereum blockchain [2].

Even though the bytecode of smart contracts is openly accessible on a public blockchain like Ethereum, it is difficult to analyze the activities, as new contracts are deployed as chunks of machine code at a rate of 10,000 and more per day. To gain insights, it would be useful to detect automatically characteristic functionalities in contracts and to tag the bytecodes according to their semantic properties. This could then be used to classify contracts by their purpose and to explore the landscape of contracts, or to search for specific types of contracts.

The detection of semantic properties is actually well established when it comes to vulnerabilities and security properties. Methods from static and dynamic program analysis as well as formal methods are employed to detect patterns characteristic of weaknesses, including symbolic execution, taint analysis, fuzzing, SMT solving and model checking. The aim of this thesis is to extend this line of research to properties beyond vulnerability detection.

2 Aim of the Thesis

The aim of this work is to establish a collection of semantic properties of Ethereum bytecode that can be identified by symbolic execution and similar methods, and that are useful for assessing the behavior of smart contracts (including vulnerabilities, but not limited to them).

We will build upon existing work on vulnerability detection and extend it to identify more general properties that capture the behavior and purpose of smart contracts. This includes concise definitions of the properties, such that they can be detected in the bytecode. We will focus on existential properties as universal properties require different approaches.

The expected contribution of this work is the development of a proof-of-concept prototype for identifying the defined properties, along with a set of definitions and guidelines for identifying semantic properties in the bytecode of smart contracts.

We address the following research questions:

- What aspects of smart contracts lend themselves to being defined as an existential property of bytecode?
- How common are these properties in smart contracts on Ethereum?

- How can the detection of semantic properties in smart contract bytecode be used to identify the purpose or behavior of a smart contract?

3 Methodological Approach

1. **Literature review:**
   The literature review will include three topics: symbolic execution with a focus on smart contracts, (mostly informal) definitions of vulnerabilities and other aspects of smart contract bytecode, as well as characteristics of the entirety of smart contracts on Ethereum.

2. **Property Definition**:
   Based on aspects of smart contracts found in the literature, we will investigate their suitability for formal definition. We will select the most promising ones to be identifiable in bytecode.

3. **Symbolic Execution:**
   In order to automate the identification of semantic properties in bytecode, a suitable symbolic execution machine is needed. Since we intend to build upon existing work, we identify and evaluate potential frameworks.

4. **Proof-of-concept prototype:**
   We select a suitable framework for symbolic execution, extend it to suit our purposes, and implement the identification of selected properties.

5. **Evaluation:**
   A set of contracts with known purpose will be used as a ground truth to evaluate the prototype. This will provide insight into the effectiveness of the approach and identify areas for improvement.

4 State of the Art

Recent years have witnessed an increased interest in methods and accompanying tools for analyzing the bytecode of smart contracts [3]. While the majority of these scientific approaches focus on potential and actual vulnerabilities in the code [4], there are some works investigating the the range of purposes or application areas that smart contracts are used for [5].

Xu et al. [6] present a collection of design patterns for blockchain-based applications, 21 in total, organized into four categories: data management, smart contract design, user interaction, and system interoperability.

Hu et al. [7] apply a transaction-based approach to identify groups of similar smart contract with the aim to detect anomalies and malicious behaviour.

Di Angelo and Salzer [8] employ semi-automatic methods to characterize several types of smart contracts on the Ethereum blockchain. They analyze a dataset of over 32 million smart contracts to identify patterns and trends in the usage of smart contracts. Employing clustering algorithms on the bytecode, they uncover groups of contracts with interesting properties.

In his master thesis, Gorgoris conducted an in depth analysis of authentication patterns in smart contracts, with the goal of detecting administrator addresses [9].

A prominent group of contracts on Ethereum are on-chain wallets since they are numerous and handle valuable assets. Di Angelo and Salzer [10, 11] provide an extensive analysis of the wallet contracts deployed on Ethereum with the aim of identifying different types of wallet contracts and to provide a comprehensive overview of their usage patterns and security profiles.

Most relevant to our work are property-based approaches, for which Tolmach et al. [12] give an recent overview. Moreover, they propose a taxonomy of smart contract properties that go beyond vulnerabilities.

The work of Schneidewind et al. [13] propose a static analyzer for the EVM bytecode, which is provably sound. It is focused on vulnerabilities caused by the interaction of different contracts (re-entrant functions). The approach relies on the property of functions/methods in a smart contract being *single entrant*.

5 Relevance to the Curriculum of the Master in Computer Engineering

The proposed work builds upon and extends the theoretical and practical knowledge and skills conveyed by the curriculum of Computer Engineering. For the work on the thesis, topics taught in the field of Formal Methods and Computer-Aided Verification serve as a basis, in particular to characterize semantic properties formally and to specify contraints for SMT solvers. Additionally, the knowledge from the field of Algorithms and Programming is useful for analyzing the compiled low-level EVM bytecode.

The following courses are related to the topics of the thesis:

- 185.291 Formal Methods in Computer Science
- 181.144 Computer Aided Verification
- 192.065 Cryptocurrencies
- 184.703 Program Analysis
- 104.345 Analysis of Algorithms
- 184.774 Automated Deduction

References

[1] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *White paper*, 3(37):2–1, 2014.

[2] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[3] Satpal Singh Kushwaha, Sandeep Joshi, Dilbag Singh, Manjit Kaur, and Heung-No Lee. Ethereum smart contract analysis tools: A systematic review. *IEEE Access*, 2022.

[4] Heidelinde Rameder, Monika di Angelo, and Gernot Salzer. Review of automated vulnerability analysis of smart contracts on ethereum. *Frontiers in Blockchain*, 5, 2022.

[5] Shafaq Naheed Khan, Faiza Loukil, Chirine Ghedira-Guegan, Elhadj Benkhelifa, and Anoud Bani-Hani. Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-peer Networking and Applications*, 14:2901–2925, 2021.

[6] Xiwei Xu, Cesare Pautasso, Liming Zhu, Qinghua Lu, and Ingo Weber. A pattern collection for blockchain-based applications. In *Proceedings of the 23rd European Conference on Pattern Languages of Programs*, pages 1–20, 2018.

[7] Teng Hu, Xiaolei Liu, Ting Chen, Xiaosong Zhang, Xiaoming Huang, Weina Niu, Jiazhong Lu, Kun Zhou, and Yuan Liu. Transaction-based classification and detection approach for ethereum smart contract. *Information Processing & Management*, 58(2):102462, 2021.

[8] Monika Di Angelo and Gernot Salzer. Characterizing types of smart contracts in the ethereum landscape. In *Financial Cryptography and Data Security: FC 2020 International Workshops, AsiaUSEC, CoDeFi, VOTING, and WTSC, Kota Kinabalu, Malaysia, February 14, 2020, Revised Selected Papers 24*, pages 389–404. Springer, 2020.

[9] Philippos Gorgoris. Identifying administrators of smart contracts from transaction data. Diploma thesis, TU Wien, Vienna, Austria, 2021.

[10] Monika di Angelo and Gernot Salzer. Wallet contracts on Ethereum – identification, types, usage, and profiles. *arXiv:2001.06909v2*, 2021.

[11] Monika Di Angelo and Gernot Salzer. Characteristics of wallet contracts on Ethereum. In *2nd Conference on Blockchain Research and Applications for Innovative Networks and Services (BRAINS'20)*. IEEE, 2020.

[12] Palina Tolmach, Yi Li, Shang-Wei Lin, Yang Liu, and Zengxiang Li. A survey of

smart contract formal specification and verification. *ACM Computing Surveys (CSUR)*, 54(7):1–38, 2021.

[13] Clara Schneidewind, Ilya Grishchenko, Markus Scherer, and Matteo Maffei. eThor: Practical and provably sound static analysis of ethereum smart contracts. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 621–640, 2020.