


Information Flow Analysis for Detecting Non-Determinism in Blockchain (Artifact)

Luca Olivieri ✉ 
University of Verona, Italy
Corvallis Srl, Padova, Italy

Luca Negrini ✉ 
Corvallis Srl, Padova, Italy

Vincenzo Arceri ✉ 
University of Parma, Italy

Fabio Tagliaferro ✉ 
CYS4 Srl, Florence, Italy

Pietro Ferrara ✉ 
Ca' Foscari University of Venice, Italy

Agostino Cortesi ✉ 
Ca' Foscari University of Venice, Italy

Fausto Spoto ✉ 
University of Verona, Italy

Abstract

A mandatory feature for blockchain software, such as smart contracts and decentralized applications, is determinism. In fact, non-deterministic behaviors do not allow blockchain nodes to reach one common consensual state or a deterministic response, which causes the blockchain to be forked, stopped, or to deny services. While domain-specific languages are deterministic by design, general-purpose languages widely used for the development of smart contracts such as Go, provide many sources of non-determinism. However, not all non-deterministic behaviours are critical. In fact, only those that affect the state or the response of the blockchain can cause problems, as other uses (for example, logging) are only observable by the node that ex-

ecutes the application and not by others. Therefore, some frameworks for blockchains, such as Hyperledger Fabric or Cosmos SDK, do not prohibit the use of non-deterministic constructs but leave the programmer the burden of ensuring that the blockchain application is deterministic. In this paper, we present a flow-based approach to detect non-deterministic vulnerabilities which could compromise the blockchain. The analysis is implemented in GoLiSA, a semantics-based static analyzer for Go applications. Our experimental results show that GoLiSA is able to detect all vulnerabilities related to non-determinism on a significant set of applications, with better results than other open-source analyzers for blockchain software written in Go.

2012 ACM Subject Classification Security and privacy → Distributed systems security; Theory of computation → Program analysis; Theory of computation → Program verification; Software and its engineering → Software notations and tools


Keywords and phrases Static Analysis, Program Verification, Non-determinism, Blockchain, Smart contracts, DApps, Go language

Digital Object Identifier 10.4230/DARTS.9.2.23

Funding *Vincenzo Arceri*: Bando di Ateneo per la ricerca 2022, founded by University of Parma, project number: MUR_DM737_2022_FIL_PROGETTI_B_ARCERI_COFIN, *Formal verification of GPLs blockchain smart contracts*

Pietro Ferrara: SERICS (PE00000014) under the NRRP MUR program funded by the EU – NGEU, iNEST-Interconnected NordEst Innovation Ecosystem funded by PNRR (Mission 4.2, Investment 1.5) NextGeneration EU – Project ID: ECS 00000043, and SPIN-2021 “Static Analysis for Data Scientists” funded by Ca’ Foscari University

Agostino Cortesi: SERICS (PE00000014) under the NRRP MUR program funded by the EU – NGEU, iNEST-Interconnected NordEst Innovation Ecosystem funded by PNRR (Mission 4.2, Investment 1.5) NextGeneration EU – Project ID: ECS 00000043, and SPIN-2021 “Ressa-Rob” funded by Ca’ Foscari University

 © Luca Olivieri, Luca Negrini, Vincenzo Arceri, Fabio Tagliaferro, Pietro Ferrara, Agostino Cortesi, and Fausto Spoto; licensed under Creative Commons License CC-BY 4.0

Dagstuhl Artifacts Series, Vol. 9, Issue 2, Artifact No. 23, pp. 23:1–23:3

 DAGSTUHL ARTIFACTS SERIES
Dagstuhl Artifacts Series
Schloss Dagstuhl – Leibniz-Zentrum für Informatik,
Dagstuhl Publishing, Germany



Related Article Luca Olivieri, Luca Negrini, Vincenzo Arceri, Fabio Tagliaferro, Pietro Ferrara, Agostino Cortesi, and Fausto Spoto, “Information Flow Analysis for Detecting Non-Determinism in Blockchain”, in 37th European Conference on Object-Oriented Programming (ECOOP 2023), LIPIcs, Vol. 263, pp. 23:1–23:25, 2023. <https://doi.org/10.4230/LIPIcs.ECOOP.2023.23>

Related Conference 37th European Conference on Object-Oriented Programming (ECOOP 2023), July 17–21, 2023, Seattle, Washington, United States

Evaluation Policy The artifact has been evaluated as described in the ECOOP 2023 Call for Artifacts and the ACM Artifact Review and Badging Policy.

1 Scope

The scope of the artifact is to reproduce the results reported in the paper *Information Flow Analysis for Detecting Non-Determinism in Blockchain* in Proceedings of the 37th European Conference on Object-Oriented Programming (ECOOP 2023). In particular, the artifact provides a complete environment for using our analyses, to evaluate them on a set of 651 real-world Hyperledger Fabric smart contracts and to compare GoLiSA against state-of-the-art static analyzers in this domain.

2 Content

The artifact comprises the following distinct components:

- Virtual Machine (Linux)
 - README file: instruction to reproduce paper results and guide for GoLiSA
 - Analyzers: GoLiSA, ChainCode Analyzer ¹ and Revive^{CC} ²
 - Benchmark: Set of smart contracts written in Go
 - Results: generated after the execution of analyzers

In the following formats:

- Virtual Machine (Linux): .OVA
 - README file: .pdf
 - Analyzers: binaries and source code
 - Benchmark: source code written in Go
 - Results:
 - * GoLiSA result: .json
 - * ChaincodeAnalyzer: .txt (structured output format not available)
 - * Revive^{CC}: .txt (structured output format not available)

The components contained in the virtual machine can be found in the following locations:

- README file: /home/artifactvm/Desktop/README.pdf
- Analyzers: /home/artifactvm/Desktop/Analyzers
- Benchmark: /home/artifactvm/Desktop/Benchmark
- Results: /home/artifactvm/Desktop/Results

¹ Available at <https://github.com/hyperledger-labs/chaincode-analyzer>

² Available at <https://github.com/sivachokkapu/revive-cc>

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). In addition, the version of GoLiSA contained in the artifact is also available at: <https://github.com/lisa-analyzer/go-lisa/tree/ecoop2023>.

4 Tested platforms

Oracle Virtual Box 7.0. The virtual machine was tested on Windows 10 64-bit and Ubuntu 20 LTS 64-bit.

5 License

The artifact is available under Creative Commons license.

6 MD5 sum of the artifact

82404092ce81342ef212de6cefb3e5e2

7 Size of the artifact

4.99 GiB

A Getting Started

1. Import the .OVA file using Oracle Virtual Box
2. Launch the imported Virtual Machine.
3. Open the file README.pdf on the Desktop and follow the instructions.

Note: if necessary the password of ubuntu user is ecoop2023