Reproducing: Inferring Crypto API Rules from Code Changes

Analyzing the taken approach, and discuss possible improvements

**Student**

Raphael Jenni

Introduction: Analyzing big code to gain information to help a developer get insights is a long-standing topic. With the help of big code, the possibilities have increased dramatically. The paper 'Inferring Crypto API Rules from Code Changes" [PZT+ 18] attempts to use big code to detect crypto API usage changes and derive rules from them. This paper aims to reproduce its results, analyze its approach and provide directions on building upon this idea.

Approach: To do that, we recap the paper's approach and the results. We then report on the findings and issues while implementing the described approach. The overall impression is that the paper is a good starting point for further research but is not applicable for a real-world application with a large codebase. The reproduced results are documented, and problems are discussed.

Conclusion: We end the paper with three rough ideas that are based on the paper's approach: (i) "Enhance Code Change Detection" where the code change detection method should be extended to handle more cases of code changes. (ii) "End-to-end Toolchain" where a toolchain is proposed to introduce machine learning to have a system that detects errors based on code changes from open source projects. (iii) "Currently-Changing Dashboard" where a dashboard is proposed that shows currently changing API usages/upgrades, vulnerability and security issue fixes, and other changes to a developer and also delivers insights on what parts of a project need to be investigated.
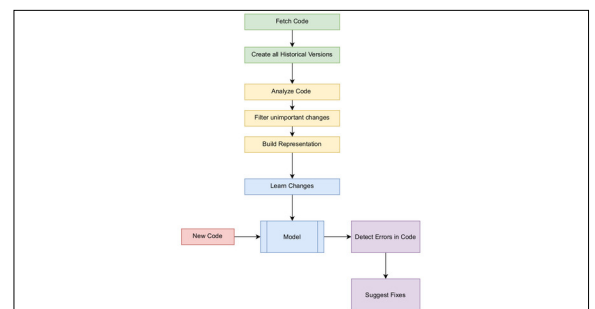
Conversion of Code into an abstract DAG
Own presentment
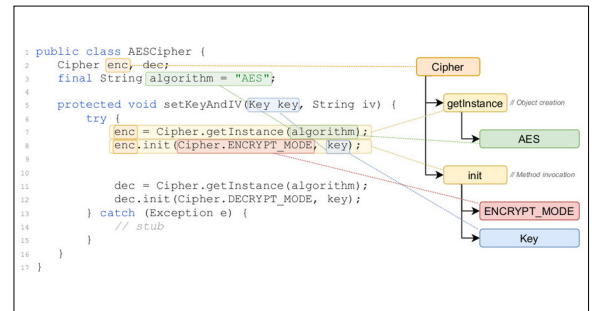
Analyzing the Code results in multiple DAGs. How to find the matching ones
Own presentment
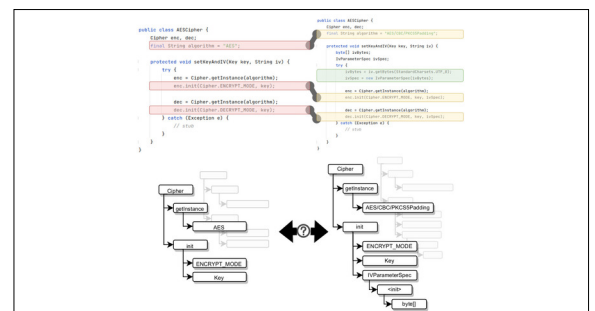
End-to-end toolchain for detecting code changes and suggesting fixes visualized
Own presentment

Examiner
Prof. Dr. Luc Bläser

Subject Area
Computer Science, Software and Systems