UNIVERSITÀ
DI PARMA

Dipartimento di Ingegneria e Architettura

Corso di Laurea in Ingegneria Informatica, Elettronica e delle Telecomunicazioni

---

Un Sistema IA per Security Testing di Smart Contract con Echidna

An AI System for Smart Contract Security Testing with Echidna

Relatore:
Prof. Michele Amoretti

Tesi di Laurea di:
Simone Orsi

Correlatori:
Ing. Stefano Cavalli

ANNO ACCADEMICO 2023/2024

Smart contracts represent a fundamental pillar of the blockchain revolution, promising to automate and secure transactions through self-executing code. However, their immutable nature, while being a strength, becomes critical when vulnerabilities are present. Once deployed, vulnerable smart contracts cannot be easily modified, making pre-deployment testing crucial.

The DeFi sector has suffered significant losses due to smart contract vulnerabilities, with recent years seeing attacks exceeding $6 billion in total losses, including the Ronin Network ($620M), Poly Network ($610M), and Wormhole ($326M) exploits. These events highlight a critical gap in smart contract security practices and the urgent need for better testing solutions.

This Thesis presents an innovative approach to address these challenges by combining artificial intelligence with established dynamic analysis tools, specifically Echidna. The developed system demonstrates that automated security testing of smart contracts can be made more accessible and cost-effective while maintaining effectiveness.

System Architecture and Implementation

The system implements the following five-stage pipeline for smart contract analysis.

1. **Input Processing:** Validates and preprocesses Solidity contracts
2. **Static Analysis:** Employs Slither to identify vulnerabilities
3. **Use Case Generation:** Utilizes AI to analyze contract structure
4. **Test Generation:** Creates comprehensive Echidna test files
5. **Dynamic Analysis:** Executes property-based fuzzing tests

The implementation leverages Python 3.8+, Anthropic's Claude API, Gradio for the web interface, and specialized testing tools. The AI integration uses Claude 3.5 Sonnet with optimized parameters for smart contract analysis.

Key Achievements

The system successfully bridges the gap between sophisticated security testing and everyday development practices by:

- Automating the generation of property-based tests through AI analysis
- Integrating static analysis tools to enhance the AI's understanding of the contract
- Providing meaningful test coverage through targeted dynamic analysis
- Delivering results in a format that developers can easily interpret

Cost-Effectiveness and Accessibility

One of the most significant advantages of the proposed system is its cost-effectiveness. Traditional smart contract audits can cost tens of thousands of dollars, making them inaccessible to many developers and small projects. The system architecture, requiring only two AI model calls per analysis, keeps operational costs extremely low. This affordability democratizes access to smart contract security testing, enabling individual developers, small projects, and startups to validate their implementations before deployment.

Limitations and Developer Responsibilities

While the system provides valuable security insights, it is important to acknowledge its limitations:

- The statistical nature of AI-driven analysis means that not all potential vulnerabilities will be identified in every run
- Developers must still review and validate the generated tests and results
- The system should be considered a complement to, rather than a replacement for, thorough security practices

Future Work

The primary direction for future enhancement lies in developing a specialized model for smart contract testing through the following activities.

- Collection of a large, labeled dataset of smart contracts and their corresponding test cases
- Fine-tuning of existing language models on this specialized dataset
- Development of evaluation metrics specific to smart contract test generation

Final Remarks

The system developed in this Thesis represents a significant step forward in making smart contract security testing more accessible to the broader development community. While it cannot completely replace professional security audits for high-stakes deployments, it provides a valuable tool for continuous security testing throughout the development process.

The demonstrated ability to generate meaningful security tests with minimal input and cost opens new possibilities for improving the overall security posture of blockchain applications. The potential for further improvement through specialized model training and dataset collection points to an exciting future where AI-driven security testing becomes increasingly sophisticated and reliable. Thus, this work not only provides immediate practical value, but also lays the groundwork for future innovations in the field of smart contract security.