**MSc. in Computing**

**Practicum Approval Form**

# Section 1: Student Details

|  |  |
| --- | --- |
| Project Title: | Smart Contract Verification Tool |
| Student ID: | 15455602 |
| Student name: | Mark Kane |
| Student email | mark.kane8@mail.dcu.ie |
| Chosen major: | Blockchain |
| Supervisor | Dr Geoff Hamilton |
| Date of Submission | 30/11/2022 |

# Section 2: About your Practicum

Please answer all questions below. Please pay special attention to the word counts in all cases.

**What is the topic of your proposed practicum? (100 words)**

The topic of the proposed practicum is smart contract verification in Blockchain. With the rapid increase in attention towards Blockchain in recent years, many sophisticated blockchain ledgers have arisen that support the storing and execution of programs on the blockchain. These programs are referred to as smart contracts, which enable digital assets to be managed and transferred on the Blockchain. However, writing smart contracts which are secure and free of defects has proven to be a very difficult process and is prone to errors. The loss and harm caused by this is compounded by the fact that smart contracts are immutable once deployed to the Blockchain. This practicum looks to solve this problem by developing a smart contract verification tool which verifies simple smart contract properties.   
  
**Please provide details of the papers you have read on this topic (details of 5 papers** expected).

1. [1] B. Dia, N. Ivaki and N. Laranjeiro, "An Empirical Evaluation of the Effectiveness of Smart Contract Verification Tools," 2021 IEEE 26th Pacific Rim International Symposium on Dependable Computing (PRDC), 2021, pp. 17-26, doi: 10.1109/PRDC53464.2021.00013.

2. [2] Dingman, Wesley et al. “Defects and Vulnerabilities in Smart Contracts, a Classification using the NIST Bugs Framework.” Int. J. Networked Distributed Comput. 7 (2019): 121-132.

3. [3] G.W. Hamilton, "Distilling Programs for Verification", Electronic Notes in Theoretical Computer Science 190(4):17-32 [c]

4. [4] N Atzei, M Bartoletti, and T Cimoli, A survey of attacks on Ethereum smart contracts (SoK), M Maffei and M Ryan (editors), Principles of Security and Trust. POST 2017. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2017, pp. 164-186. [d]

5. [5] Sharma, T., Zhou, Z., Miller, A., & Wang, Y. (2022). Exploring Security Practices of Smart Contract Developers. arXiv preprint arXiv:2204.11193. [e]  
  
**How does your proposal relate to existing work on this topic described in these papers?** (200 words)

[1] offers an empirical evaluation of the effectiveness of the smart contract verification tools on the market as recent as September 2021. In [1] they define a smart contract defect classification scheme and then use existing datasets of smart contracts written in Solidity to analyse the effectiveness of the latest versions of the state-of-the-art smart contract verification tools. [1] find that these tools have relatively low detection capabilities demonstrating a need for better verification tools.

[2] highlights the lack of a structured analysis and classification of defects/vulnerabilities in smart contracts and aims to provide this, noting that the vulnerabilities in this environment make for perfect target for financially motivated attackers. [2] also importantly highlights that unlike more classical distributed systems where bugs can be patched once detected, smart contracts are immutable and irreversible once deployed on the blockchain.

[4] provides a taxonomy of common programming pitfalls which may lead to vulnerabilities and shows how these may be exploited by attacks. [4] also notes that most of the vulnerabilities seem to be caused by a disconnect between the intuition of programmers and semantics of the language. This is reinforced by [5] finding that the code review process used by many smart contract developers is not sufficient to detect security vulnerabilities.

As is described in the section below section, the practicum proposed aims to apply the program transformation algorithm proposed by [3] to the smart contract domain.

**What are the research questions that you will attempt to answer? (200 words)**

The research question being addressed by this practicum is clearly defined: can the program transformation algorithm called *distillation* proposed by [3] be used to facilitate the verification of smart contracts on the Cardano blockchain. Although initially designed to be used for the optimization of programs, [3] argues that it transforms programs into a specialized form that makes the properties of said program easily verified through the application of inductive proof rules. The distillation algorithm has already been implemented and incorporated into an automatic inductive theorem prover named Poitín, and has been implemented in the Haskell programming language [3]. Therefore, I believe that it is a suitable candidate for application to smart contracts. This practicum will aim to create a simple tool which accepts smart contracts and utilizes this distillation algorithm to verify properties of the smart contract and detect any vulnerabilities that exist. It will then evaluate whether *distillation* is an effective tool for smart contract verification by testing the tool on a diverse set of smart contracts. This practicum will also aim to provide a publicly available dataset for smart contracts written in Haskell with a range of common smart contract vulnerabilities (should there be none already available today) to support further research in this area.

**How will you explore these questions? (Please address the following points. Note that three or four sentences on each will suffice.)**   
  
**- What software and programming environment will you use?**

This practicum will be implemented on the Cardano blockchain using the Plutus smart contract platform which uses a subset of Haskell as its programming language.

**- What coding/development will you do?**

I will explore applying the distillation program transformation algorithm to smart contracts written in Haskell and evaluate its effectiveness to facilitate the verification of properties of the smart contract. I will look to develop a simple tool which will accept and verify smart contracts before they are deployed on the blockchain.

**- What data will be used for your investigations?**

A dataset of smart contracts with different known types of vulnerabilities.

**- Is this data currently available, it not, where will it come from?**

Ideally, I would like to find a premade dataset. There are such datasets available written in Solidity for the Ethereum blockchain like those used in [1], however, it is difficult to find such a dataset for Cardano smart contracts written using Plutus. Failing finding a premade dataset, I will create my own.

**- What experiments do you expect to run?**

I expect to test the smart contract verification tool on a wide range of smart contracts with different vulnerabilities.

**- What output do you expect to gather?**

I expect to get an output indicating the accuracy of the tool in detecting smart contract vulnerabilities.

**- How will the results be evaluated?**

The results will be evaluated by testing the smart contract verification tool on a labelled dataset (smart contracts labelled as being with or without vulnerabilities) to get a measure of the accuracy of the tool. There is not much work done in this area for the Cardano blockchain specifically, but comparisons can be drawn with results of smart contract verification tools used on the more mainstream Ethereum blockchain.