Project Definition Document (PDD)

Cover Sheet

Decompiling Ethereum EVM ByteCode for Static Analysis

Prepared for: City UOL, Department of Computer Science
Prepared by: Gera Jahja, Computer Science Bsc
Email: gera.jahja@city.ac.uk / g_jahja31@outlook.com
Consultant: Martin Nyx Brain
Academic Client: Martin Nyx Brain and Michał Król
Date: February 2022
This project is an academic proposal supervised by Martin Nyx Brain and Michał Król. The project will be to write a decompiler so that it can convert EVM byte code into C that can be handled by the CPROVER tools.
Word count:1280 (not including tables, diagrams and refferencing section)

Project Proposal

Problem to be solved:

Ethereum is one of the most exciting block chain technologies as it is not just a cryptocurrency but also supports smart-contracts. These are programs that are written in a variety of programming languages and compiled to EVM, a byte-code format similar to JVM, before they are run on the block chain. The security of these contracts is vital as they can control significant amounts of cryptocurrency.

This project requires me to translate EVM byte code to C code. The purpose of this translation is to see whether we can detect bugs using CPROVER tools. If we can successfully use tools used to detect C code bugs on EVM this means we can add verification to EVM (as well as aiding our understanding of the behaviour of the smart contracts) and ensure that the Ethereum currency that is associated with the byte code is protected and less viable to hacking. Decompiling is a part of reverse engineering, I will be using this approach to convert EVM byte code to op code, and then generating C code from this.

When looking to verify whether the software works I will be specifically be looking at :

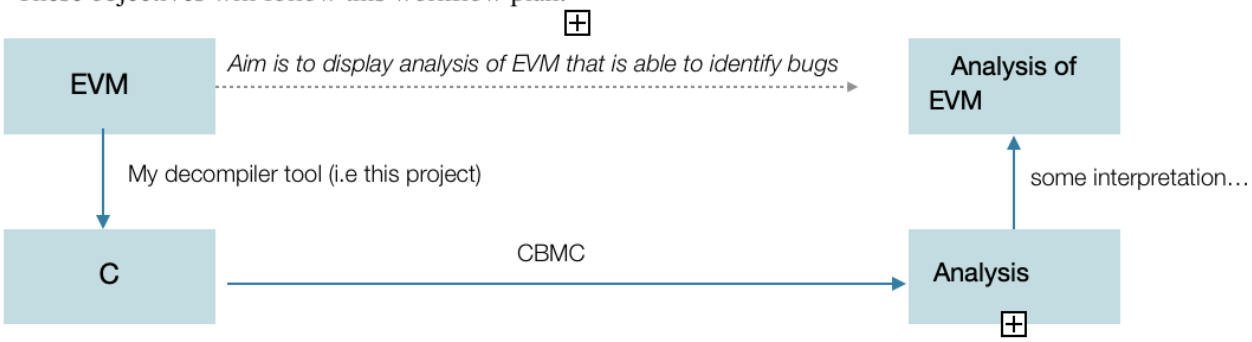
- Will the program crash? Can it be hacked? Can we do it without running the program? (Static Analysis, i.e using CPROVER)

The Verification tool for C is CBMC and there are some verification tools for EVM or Solidity, etc.

Project Objectives:

-Produce a subset of EVM byte code that has been decompiled to C code.
-Develop a program that displays this decompilation
-Ensure this subset (of EVM byte code)can be run through the CPROVER tool CBMC
-Investigate to what degree do these tools allow us to analyse the EVM contracts , I.e is it able to aid us in detecting bugs?

These objectives will follow this workflow plan:



Project Beneficiaries:

Beneficiaries of the project include:

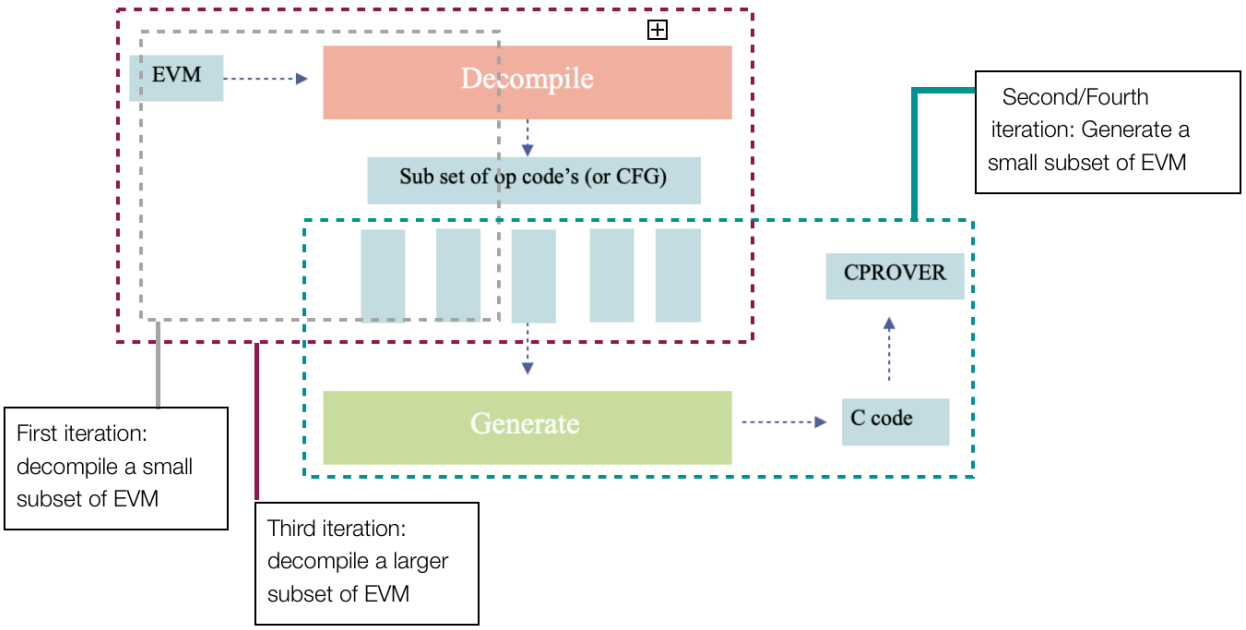
- My clients and I-Martin is quite interested in the verification end of the problem we are trying to solve, while Michael is interested in the Ethereum cryptocurrency side of things. My aim is to pave a connection between the two.

- Academics - The findings of this development may be of interests to academics in similar fields of work, such as cyber security, decompilation and more.

- People using blockchain - If we are able to validate smart contracts it can prevent huge financial losses for people that are part of the Ethereum blockchain. In the past there have been times where hacking into the currency has lead to cryptocurrency being destroyed. In 2017, “$300m of cryptocurrency was lost after a series of bugs in a popular digital wallet service led one curious developer to accidentally take control of and then lock up the funds” Locating these bugs can prevent similar cases from occurring, adding more security to the currency and more protection for people using block chain.

- In general, building tools that are useful is the main benefit of this project. Finding and ideally preventing bugs with evidence that it works would be the ideal outcome of this project.

Proposed Project structure (decided with my academic clients during a meeting on 03/02/2022) for my Main Product (To be developed using java):

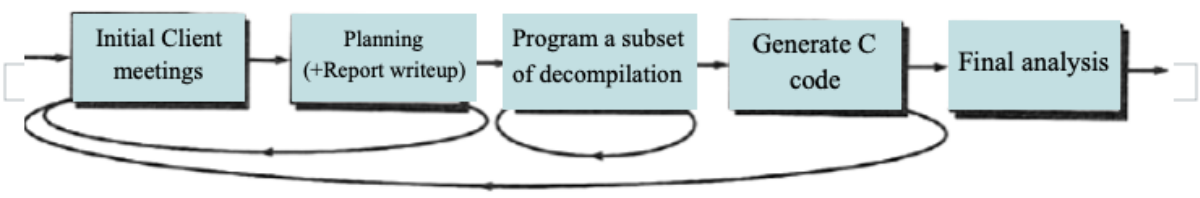


Iterative Development:

I will be splitting my project into three builds and will have **numbered objectives** for each:

1. **Minimum Viable Product** : Complete by mid-March :
   - Research all existing implementations of similar projects (2-3 days)
   - Write-up report (3-4 days)
   - The program will take some EVM code and decompile it , translate into a list of Op code (or a CGF) (2+ weeks, depending on whether the development goes smoothly)
   - Get client feedback, test that this product has no syntax or logical errors. (2-3 hours)
2. **The Main Product:** functional but with no extra subsets of op code : Complete by mid-April :
   - Extend Report with updates (3-4 days)
   - The program will take the decompiled Op code and generate reliable C code AND will successfully be usable by CPROVER tests(3+weeks)
   - Get client feedback, test that this product has no syntax or logical errors. (2-3 hours)
3. **Additional Features:** Complete towards the start of May :
   - Look at the behaviour of the decompiled C program and add additional features based on the main product's current output(1+week(s))
   - Add additional op code features (extending the decompiler and code generator made in the first two iterations) (2+ weeks)

Refer to this iterative workflow diagram(each stage must go through testing or a client meeting before moving on to the next step, and preferably each iteration SHOULD take 3 weeks (4 max)):



Risks:

Project Management:

- The EVM byte code may be difficult to decompile into C, and may not produce perfect C code. This should not be an issue if my prior research is well done at the start of the project. This research will be continued throughout the project incase I discover more up to date examples.

- What the CPROVER identifies as a bug may not be a bug for the EVM code… is this an accurate assumption to assume C code bugs are the same as EVM bugs?
  - To refrain this from preventing the success of my project I will be working using an iterative approach, so if the first iterations fail I will have enough time to plan a new method.

Technical:

- What if the systems don't work , I.e platforms and breaking of tools?
  - I will be displaying mitigation and proof of my attempts for the solutions , I am also prepared to use virtual machines , (for windows dependencies as I am using a MacBook)
- Am I able to produce a program to satisfy the Academic Client?
  - Regular Client meetings every week will prevent this from happening

Personal:

- There may be Covid restrictions that prevent some meetings with the client. To prevent this I am prepared to communicate using online meetings if the situation calls for it.

- Will the detection of bugs possibly be incorrect?
  - As we are dealing with cryptocurrency the risks of being responsible for the authentication of smart contracts could cause losses to people relying on the tool if it is used and is incorrect.

- How will the tool be used? And what for?
  - This project could be an example of dual use technology, while I am aware of possible concerning uses of this type of tool , if I find bugs I will not be using my findings in a malicious manner.

  - If we find a bug this could be ethically wrong if it's used for personal gain. If the tool is used as a form of Responsible disclosure then its ethical. However if the tool is used unethically then knowledge of bugs in EVM programs, that are supposed to be safe, could put a lot of smart contracts in danger.

Risk assessment:

Likelihood (of this being a risk) : 1-5 (1 being very unlikely, 5 being very likely)
Severity(how necessary this is) : 1-5 (1 being not impactful, 5 being very impactful)
Score is the Likelihood*Severity, (0-6 is low risk, 6-14 is medium risk, 15+ is high risk)

Objective	Likelihood	Severity	Score	Risk	Prevention
The program will take some EVM code and decompile it , translate into some sort of reliable C code.	1	5	5	Problems with incorrectly decompiling due to lack of knowledge of Solidity and EVM	Look at existing tools or extend an existing decompiler such as https://github.com/MrLuit/evm and get comfortable with EVM and solidity via tutorials and existing smart contracts.
The program will take the decompiled Op code and generate reliable C code	3	5	15	Unable to convert the op code to reliable C	Testing throughout the implementation with the client will prevent this problem
The generated C code will successfully be usable by CPROVER tests	2	5	10	CPROVER tests cannot be applied to the generated code	Re-attempt the second phase and look at existing code translation generators
Look at the behaviour of the decompiled C program and add additional features based on the main product's current output	2	2	4	possibly the C code generated could be incorrect	Look at existing op code to C conversions to identify where the program has gone wrong
Add additional op code features (extending the decompiler and code generator made in the first two iterations)	5	3	15	The additional Op code features may be too complex to decompile and then generate C code from	Tackle easier commands first and prioritise the generation of valid C code over the quantity of op codes decompiles

References:

The Guardian. (2017). *"$300m in cryptocurrency" accidentally lost forever due to bug*. [online] Available at: https://www.theguardian.com/technology/2017/nov/08/cryptocurrency-300m-dollars-stolen-bug-ether [Accessed 3 Feb. 2022].

Research Ethics Checklist

CSREC –Review Form – Part A : Ethics Checklist *Version 4.4, October 2015, April 2019*

Research Ethics Review Form: BSc, MSc and MA Projects

Computer Science Research Ethics Committee (CSREC)

http://www.city.ac.uk/department-computer-science/research-ethics

Undergraduate and postgraduate students undertaking their final project in the Department of Computer Science are required to consider the ethics of their project work and to ensure that it complies with research ethics guidelines. In some cases, a project will need approval from an ethics committee before it can proceed. Usually, but not always, this will be because the student is involving other people ("participants") in the project.

In order to ensure that appropriate consideration is given to ethical issues, all students must complete this form and attach it to their project proposal document. There are two parts:

PART A: Ethics Checklist. All students must complete this part. The checklist identifies whether the project requires ethical approval and, if so, where to apply for approval.

PART B: Ethics Proportionate Review Form. Students who have answered "no" to all questions in A1, A2 and A3 and "yes" to question 4 in A4 in the ethics checklist must complete this part. The project supervisor has delegated authority to provide approval in such cases that are considered to involve MINIMAL risk. The approval may be ***provisional*** – *identifying the planned research as* likely to involve MINIMAL RISK. In such cases you must additionally seek ***full approval*** from the supervisor as the project progresses and details are established. ***Full approval*** must be acquired in writing, before beginning the planned research.

A.1 If you answer YES to any of the questions in this block, you must apply to an appropriate external ethics committee for approval and log this approval as an External Application through Research Ethics Online - https://ethics.city.ac.uk/		*Delete as appropriate*
1.1	Does your research require approval from the National Research Ethics Service (NRES)? *e.g. because you are recruiting current NHS patients or staff?* *If you are unsure try - https://www.hra.nhs.uk/approvals-amendments/what-approvals-do-i-need/*	**NO**
1.2	Will you recruit participants who fall under the auspices of the Mental Capacity Act? *Such research needs to be approved by an external ethics committee such as NRES or the Social Care Research Ethics Committee - http://www.scie.org.uk/research/ethics-committee/*	**NO**
1.3	Will you recruit any participants who are currently under the auspices of the Criminal Justice System, for example, but not limited to, people on remand, prisoners and those on probation? *Such research needs to be authorised by the ethics approval system of the National Offender Management Service.*	**NO**
A.2 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee, you must apply for approval from the Senate Research Ethics Committee (SREC) through Research Ethics Online - https://ethics.city.ac.uk/		*Delete as appropriate*
2.1	Does your research involve participants who are unable to give informed consent? *For example, but not limited to, people who may have a degree of learning disability or mental health problem, that means they are unable to make an informed decision on their own behalf.*	**NO**
2.2	Is there a risk that your research might lead to disclosures from participants concerning their involvement in illegal activities?	**NO**
2.3	Is there a risk that obscene and or illegal material may need to be accessed for your research study (including online content and other material)?	**NO**

2.4	Does your project involve participants disclosing information about special category or sensitive subjects? *For example, but not limited to: racial or ethnic origin; political opinions; religious beliefs; trade union membership; physical or mental health; sexual life; criminal offences and proceedings*	**NO**
2.5	Does your research involve you travelling to another country outside of the UK, where the Foreign & Commonwealth Office has issued a travel warning that affects the area in which you will study? *Please check the latest guidance from the FCO - http://www.fco.gov.uk/en/*	**NO**
2.6	Does your research involve invasive or intrusive procedures? *These may include, but are not limited to, electrical stimulation, heat, cold or bruising.*	**NO**
2.7	Does your research involve animals?	**NO**
2.8	Does your research involve the administration of drugs, placebos or other substances to study participants?	**NO**
A.3 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee or the SREC, you must apply for approval from the Computer Science Research Ethics Committee (CSREC) through Research Ethics Online - https://ethics.city.ac.uk/ **Depending on the level of risk associated with your application, it may be referred to the Senate Research Ethics Committee.**		*Delete as appropriate*
3.1	Does your research involve participants who are under the age of 18?	**NO**
3.2	Does your research involve adults who are vulnerable because of their social, psychological or medical circumstances (vulnerable adults)? *This includes adults with cognitive and / or learning disabilities, adults with physical disabilities and older people.*	**NO**
3.3	Are participants recruited because they are staff or students of City, University of London? *For example, students studying on a particular course or module.* *If yes, then approval is also required from the Head of Department or Programme Director.*	**NO**
3.4	Does your research involve intentional deception of participants?	**NO**
3.5	Does your research involve participants taking part without their informed consent?	**NO**
3.5	Is the risk posed to participants greater than that in normal working life?	**NO**
3.7	Is the risk posed to you, the researcher(s), greater than that in normal working life?	**NO**
A.4 If you answer YES to the following question and your answers to all other questions in sections A1, A2 and A3 are NO, then your project is deemed to be of MINIMAL RISK. **If this is the case, then you can apply for approval through your supervisor under PROPORTIONATE REVIEW. You do so by completing PART B of this form.** **If you have answered NO to all questions on this form, then your project does not require ethical approval. You should submit and retain this form as evidence of this.**		*Delete as appropriate*
4	Does your project involve human participants or their identifiable personal data? *For example, as interviewees, respondents to a survey or participants in testing.*	**NO**