Principles of Decentralized Ledgers
Coursework
Smart Contract Based Application

Dr. Arthur Gervais

2019

1 Introduction

In this coursework, you are expected to become familiar with the advanced development of an EVM (Ethereum Virtual Machine) based smart contract.

You're free to write the smart contracts in any language, but I recommend using Solidity. We are discussing Solidity within the course, and the Achievement.Network will help you to get started from the basics. Alternatively, you can also follow `https://solidity.readthedocs.io/en/develop/solidity-in-depth.html`.

2 Schedule and Guidelines

The coursework is scheduled according to the following guidelines.

- Pick a subject
- The project is assessed individually
- Implement the application
- Write a short report (max 4 pages)
- Report + Code due by 13th of February 2018
- You can present your work at the end of the course if you wish so

You're free to choose one of the following topics (cf. Section 3), or a topic of your interest. If you plan to pursue your own topic, please do speak briefly to me in advance.

3 Topics

- King of Ether (+)
- Battleship (++)
- Micropayment Channel (+++)
- Rock Paper Scissors (++)
- Magic: The Gathering (+++)
- Choose your own topic (?)

Difficulty: Easy (+), Medium (++), Advanced (+++)

3.1 King of Ether

This game should allow a user to claim being a *king of Ether* by paying an amount of Ether asked by the current king. The current king receives a financial reward from the difference between the price he paid to the previous king. When a user claims the throne, the contract sends the compensation to the ceded king, and assigns the user as the new king.

(Optional +++) Extend it with a hidden auction that runs over a given timeframe.

3.2 Battleship

Battleship is a guessing game for two players. It is played on ruled grids on which the players' fleets of ships (including battleships) are marked. The locations of the fleet are concealed from the other player. Players alternate turns calling "shots" at the other player's ships, and the objective of the game is to destroy the opposing player's fleet.

You're asked to implement the game logic so that 2 player can register to a game. You can make the simplifying assumption that the adversary is not able to read the blockchain, i.e. the adversary is not able to view the location of the ships of the opponent.

3.3 Micropayment Channel

A Micropayment Channel or Payment Channel is a technique designed to allow users to make multiple transactions without committing all of the transactions to the blockchain. In a typical payment channel, only two transactions are added to the blockchain but an unlimited or nearly unlimited number of payments can be made between the participants.

Participants can then effectively perform off-chain transactions, that are external to the blockchain, but still secured by the blockchain.

For further information, I recommend reading `https://arxiv.org/pdf/1710.02964.pdf` and `https://eprint.iacr.org/2017/823.pdf`.

You are expected to implement at least a uni-directional payment channel.

3.4 Rock Paper Scissors

The Rock Paper Scissors game is a traditional two-party game. The goals are:

- Allow two player to join the game.
- Each player chooses one of rock, paper, scissor.
- The player reveals their choices.
- The winner gets a financial reward.

The smart contract needs to be resistant against an adversary that tries to attack the game and is able to read the blockchain's content.

Over time, the expected gain for an honest party should be at least equal.

3.5 Magic: The Gathering

CryptoKitties allows players to exchange cryptocollectibles and breed new kitties. In this exercise you can implement your own game, that is for instance card-based.

The task is therefore to design a similar game as cryptokitties, but where users can sell their cards instead of kitties.

Battle logic (i.e. fights between cards) is optional, the focus of this work should be on trading, sharing and ownership logic that is handled by the smart contract.

4 Grading

The grading will be mostly influenced by

- the difficulty of the chosen topic
- the quality of the code. Simplicity is preferred over complexity.
- secure programming considerations are a plus
- building a web interface is optional but a big plus

The report should contain the following information:

1. Topic description and what's your architectural choice on how to implement it (application logic description).
2. Threat model description, i.e. what an adversary is supposed to be able to do.

3. At least one screenshot showcasing how you call your contract (with a transaction), and what the return result is. A screenshot of the web interface is a big plus.

5 Conclusion

This exercise should expose you to the art of writing and interacting with a smart contract. Smart contracts are very different to traditional programs, have the ability to handle monetary amounts and are extremely costly to execute.