Durham University

Computer Science

Summative Assignment

Module code and title	COMP4137 Blockchain and Cryptocurrencies
Academic year	2022/23
Submodule title	-
Coursework title	Blockchain and Cryptocurrencies Coursework
Coursework credits	5 credits
Lecturer	Gagangeet Aujla
Deadline*	Thursday, January 19, 2023 14:00
Hand in method	Ultra

Additional coursework files	*Paper1.pdf, paper2.pdf, paper3.pdf*
Required submission items and formats	*parta.py, partb.sol, report.pdf*

*** This is the deadline for all submissions except where an approved extension is in place. Late submissions received within 5 working days of the deadline will be capped at 40%. Late submissions received later than 5 days after the deadline will received a mark of 0.**

COMP4137 Assignment

Design and Implementation of an Agro Supply Chain Scenario using Blockchain (100 marks)

Blockchain technology is not only limited to financial technologies but has captured a wide range of modern-day applications. So, it is important to understand how blockchain can go beyond financial transactions and be used for non-financial purposes. In this imaginary supply chain scenario, you are hired as a blockchain expert for a newly established agro-organisation. Your role is to perform technical analysis based on the organisation's requirements, design the solution and implement it accordingly. This organisation would like to analyse a scenario where they want to control and coordinate their agro-supply chain to keep their inventory and money transfers transparent. Thus, they plan to use blockchain technology. Now, your task is to provide them with a small-scale use case that can help them understand how blockchain technology can make their supply chain secure and transparent.

Here, you are required to propose a solution for some of the tasks. You should critically analyse the problem, consider possible design choices and justify your solution. You should also include the **implementation details (including the code files)** for these tasks and **a report** so that agro-organisation people can understand your solution.

1. **Design Tasks:** To realise the imaginary supply chain scenario stated above, you are required to accomplish the following design tasks.
   a) Identify appropriate stakeholders in the agro-supply chain and explain their role and associated tasks they will undertake. **(5 marks)**
   (You can refer to this article for more understanding of supply chain and blockchain: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9201315). The pdf of this paper is also available in the 'reading material' on the ULTRA.
   b) Design the interactions between at least four supply chain stakeholders identified in task 1(a) and represent the interactions in the form of a sequence diagram. These transactions must include inventory supply (material or items) and financial transactions initiated by supply chain stakeholders. (e.g., if a farmer cultivates crops (raw material) and sells them to a food processing manufacturer, then what data, inventory and financial interactions will occur between these two stakeholders). **(5 marks)**
   (You can refer to this article for more understanding of the interactions in the supply chain: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9680497 and https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8946187). The pdf of this paper is also available in the 'reading material' on the ULTRA.

2. **DApp Tasks:** The next task is to associate the identified supply chain stakeholders with identities that can be verified and traced back. You are required to accomplish the following implementation tasks. **(10 marks)**
   a) Generate the pseudo-random identities for stakeholders (at least four) in the supply chain process (let's say A, B, C and D) and register them on the blockchain. The generated random identities must include:
      - a private ID, i.e., a privateKey
        (eg., L3qambQufogBWYgCwMvGafwi7PskT3asj4hHG7uPH9YLZEvQiidQ)
      - a digital wallet address

(eg., 15sZo91rVRFSpabb3vAbUx9esiGNPRpU5s)

- As the generated hash ID is complex, some QR code (like) should be generated.

(**Hint:** To make the task interactive and easy to deploy, you can represent it as a webpage)

You may need to set up the environment with the help of the components enlisted below.

- Get the coinables/buidljs libraries into your system from https://github.com/coinables/buidljs
- For the generation of the QR codes, you can get the library from https://github.com/davidshimjs/qrcodejs/blob/master/qrcode.js

3. **Blockchain Tasks:** Implement the interaction diagram (using python) to perform the following tasks. **(15 marks)**

   a) Generate a genesis block that must include the following attributes and perform some tasks.
   - Block ID (Hint: The very first block is the genesis block)
   - Timestamp
   - Transaction count
   - Previous block hash (Hint: For genesis block, this field will be null)
   - Nonce (Hint: Use a dummy nonce as mining has not been performed yet)
   - A transaction in the genesis block.
   - Compute the hash of the block. (Hint: use hashlib library in python3 and SHA256 hashing.)

   b) Create another block and link with the previous block using a hash pointer. In this block, you must include the transactions initiated in the interaction diagram, including all the blockchain attributes:
   - Add a few sample transactions initiated in the sequence diagram designed in 1(b).
   - Add the previous block hash
   - Find the valid nonce value (target hash leading five zeros) which makes the block valid.
   - Compute the hash of the block.

   [**Hint:** All the transactions should be signed, and you should follow all the tenants of the transaction details (like signature checking)]

   - For Python 3, e.g., pycharm community edition can be downloaded from https://www.jetbrains.com/pycharm/download (You may use python 3 modules hashlib, ecdsa, json and datetime). You can use additional libraries if required (e.g., for hex conversion).

4. **Mining Tasks:** Use Proof of Work (PoW) to mine the valid block and perform the following analysis based on difficulty and processing time. **(15 marks)**

   a) When executed on your computer, analyse the impact of an increase in difficulty (*hint: number of leading zeros in target hash*) on the processing time. (Start with difficulty as "0" leading zeros and increase it to at least "10" leading zeros). If your machine fails to compute a nonce within 60 minutes time limit (set this limit inside your code), then the output should be, "It's very difficult to find nonce") You must state your personal computer computational capabilities. (plot the outcome)

   b) Take any given length of difficulty and find the average number of brute force attempts required to find the nonce. (Plot the variation of number of brute force attempts with respect to the "length of difficulty).

   c) Analyse the impact of an increase in difficulty on energy consumption and plot it.

5. **Traceback and Verification Tasks:** One of the key requirements of the supply chain is to trace back the transactions and verify the ownership of the transactions. You need to create an

automated system where the agro-industry managers can easily trace back the transactions. In this process, the following functionality is required. **(15 marks)**

a) Trace a specific transaction by its attributes (like transaction ID, time/date, product) and provide the time taken to trace the transaction.

b) Verify the ownership of the transaction based on the digital identities of the stakeholders and provide the time taken to trace the transaction.

I should be able to use this functionality and look into some of the transactions and check who initiated the transactions, the time of the transactions, and other relevant details. I suggest not including any sensitive data or personal identification of any party.

6. **Deploy Ethereum Smart Contracts (Recycle and Reuse):** The conventional supply chain process is linear (produce, use and discard). However, these days circular supply chain is very popular as it adds another stage (recycle and reuse) to the life cycle of the supply chain. Now, complete the following tasks based on your understanding.

a) Set up your machine to use the ethereum test network. Explain the process and significance of each step followed to set up your machine. (Hint: You can use the metmask extension on google chrome to create a metamask wallet on your browser). You have to configure your system to use the Ropsten network and do the following task. Get some test Ethers and add them to your wallet. **(5 marks)**

b) Design a smart contract for a retail store to ensure that the products can be added to the inventory, transferred to another store, verified for their shelf life, and any legitimate person can check the inventory (i.e., stocks, item, date it was added to the shelf, date it will expire, owner of the transactions, etc.). Deploy an automated smart contract that identifies the product, its shelf life, and expiry status, and initiates the 'recycle or reuse' process as appropriate for a specific product. **(15 marks)**

c) Implement these smart contracts through remix IDE (http://remix.ethereum.org/), compile and deploy them on Ganache (https://www.trufflesuite.com/ganache) and Ropsten Testnet Explorer (https://ropsten.etherscan.io). You have to provide the cost of the smart contracts in terms of gas consumption and ether (wei) for both Ganache and Ropsten. **(5marks)** Any details for energy consumption comparison will attract additional marks **(2 marks).**

Submission Requirements

Students are expected to work on the coursework individually. This assignment consists of six tasks. These tasks have to be completed and submitted as a report and code. You should submit the following code files and a single pdf report (covering all tasks) through ULTRA.

- One piece of code as a **part1.py** file, clearly differentiating the tasks. Please make your code clean and clear, and easy to follow, with comments as necessary.
- One piece of code as a **part2.sol** file. Please make your code clean and clear, and easy to follow, with comments as necessary.
- A report covering the following points **(do not exceed 2000 words)** **(8 marks).**
  - information asked in all the tasks (like random IDs, a digital address, hash ID, block and transaction details, difficulty and nonce).
  - a discussion on **Design Tasks, DApp Tasks, Blockchain Tasks, Mining Tasks, Traceback and Verification Tasks, Deploy Ethereum Smart Contracts (Recycle and Reuse).** This should include the graphs, plot, and snapshot as and when required.

Please make it easy to find the above elements in the report.

If any task is not provided in the report then they will get partial score based on the code.

Frequently Asked Questions:

What the examiners expect from program implementation:

- Your program must be runnable – a program that partially works or does not run at all will receive no mark.
- Your source code should be documented with comments.
- Apart from performing the requested functionality, your design should aim at a clear programming logic. Your proposed solution should also be as robust as possible.

What the examiners expect from the report:

- Your report should explain your solution with reference to your source code. You are NOT encouraged to copy the whole source code to your report, but you may refer to/quote important lines if you believe that is helpful.
- If there are any features that you wish to highlight, you are also encouraged to do so such that your examiner can pay attention to them.
- You should also provide support and justification for your design.

Marking

The marks are allocated to the sub-tasks of the code part and the report. If multiple sub-tasks are under one task, then the marks will be divided equally among all sub-tasks. I may also look at the code to allocate partial credit if some of the tasks in the code are incorrect or not executable. If the code is not clear and easy for me to understand, I may not be able to award you the marks you might otherwise have got. The report only needs to contain the requested elements; you must be specific to the questions and not exceed the word count.

Word Limit policy

The report word count will:

- Include all the text, including title, preface, introduction, in-text citations, quotations, footnotes, and any other item not specifically excluded below.
- Exclude diagrams, tables (including tables/lists of contents and figures), equations, executive summary/abstract, acknowledgements, declaration, bibliography/list of references, and appendices. However, it is not appropriate to use diagrams or tables merely as a way of circumventing the word limit. If a student uses a table or figure as a means of presenting his/her own words, then this is included in the word count.

Plagiarism and collusion

Your assignment will be put through the plagiarism detection service on the Ultra. Students suspected of plagiarism, either of published work or work from unpublished sources, including the work of other students, or of collusion, will be dealt with according to the University guidelines.

https://www.dur.ac.uk/learningandteaching.handbook/6/2/4/

You are not allowed to do the following

- Do not involve in plagiarism and collusion.
- Do not use online plagiarism checking tools as they may save your report in their repository leading to complications when you submit your report on ULTRA.
- Do not exceed the word limit in the report.
- Do not submit separate reports for different tasks (you should submit a single report only).
- Don’t use figures or snapshots merely to circumvent the word limit.