Durham University

COMP4137 Assignment

Module/Lecture Course:	Blockchain and Cryptocurrencies
Deadline for submission:	03/02/2022
Deadline for marks and feedback to be returned to students:	04/03/2022
Submission instructions:	Submit all files via Blackboard Ultra
Submission file type(s) required:	You should submit separate code files and a single PDF document for the individual reports (2000 words maximum).
Format:	Report as a PDF document and code files for the individual tasks. Do not put your name on your report, just your username.
Contribution:	The report and code files contributes 100% to the final mark for the module.

Requirements

Students are expected to work on the coursework individually. This assignment consists of four tasks. The first three taks include a combination of {coding/script and report} but the fourth task requires a written report only. Full details follow but, in short, the report should contain a section for each of the four tasks which follow and include:

- Task 1: Blockchain Design and Implementation for a given Scenario **{code + report} [35 marks]**;
- Task 2: Interacting with Bitcoin Blockchain and Scripting Language **{script + report} [20 marks]**;
- Task 3: Ethereum Smart Contracts **{code + report} [30 marks];** and
- Task 4: Cryptocurrency of the future **{report} [15 marks].**

You should submit the following code files and a single pdf report (covering all four tasks) through ULTRA. Please don't zip files together or rename them - just attach them to the ULTRA submission.

- task1.py
- task2.txt
- task3.sol
- report.pdf (including all four tasks and label each task separately)

Task 1. Blockchain Design and Implementation for a given Scenario (35 Marks)

In this imaginary scenario, you are a security expert for a newly established financial investment organisation. Your role is to perform technical analysis based on the organisations requirements, design the solution and implement it accordingly. Now, this organisation would like to analyze a scenario where three parties (lets say A, B and C) want to initiate a business venture. They want to keep their money transfers transparent and thus plan to use blockchain technology (like bitcoin blockchain). Now, your task is to provide them with a small scale protoype that can help them understand how blockchain technology can make their transactions secure and transaparent.

Here, you are required to propose a written 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 task and a report** such that business organisation people would be able to understand your solution.

1. **DApp Tasks:** In order to realise the imaginary scenario stated above, you are required to accomplish the following implementation tasks.
   - Generate the pseudo-random identities for three parties (lets say A, B and C) that want to initiate the business venture. **(2 marks)**
   - The generated random identities must include **(5 marks)**:
     - a private ID, i.e., a privateKey (eg., 15sZo91rVRFSpabb3vAbUx9esiGNPRpU5s)
     - a digital wallet address should be generated by hashing the private ID (eg., L3qambQufogBWYgCwMvGafwi7PskT3asj4hHG7uPH9YLZEvQiidQ)
     - As the generated hash ID is complex, so a QR code (like) should be generated.

   (**Hint:** You can use *coinables/buidljs* library. To make the task interactive and easy to deploy, you can consider it in form of a webpage, i.e., HTML. This used to display the above process as a webpage. You donot need to provide the code for this but you can add the snapshot of the webpage in your report). **(3marks)**

2. **Blockchain Tasks:** Now, the next task is to generate a genesis block considering the identities generated above. In this block, you must include the following attributes. (Hint: You can use python for the this task). **(15 marks)**
   - Block ID (Hint: The very first block is the genesis block)
   - Add the previous block hash (Hint: For genesis block this field with be null)
   - Nonce (Hint: Use a dummy nonce as mining has not been performed yet)
   - Add a coin creation transaction in the genesis block.
   - Compute the hash of the block. (Hint: use hashlib library in python3 and SHA256 hashing.)

   Create another block and use a hash pointer to link with the previous block. In this block, you must include the following attributes:
   - Block ID
   - Add a few sample transactions initiated by three parties (i.e., A, B , and C) in the block (consider at least 3 transactions).
   - Add the previous block hash
   - Find the valid nonce value (target hash leading 5 zero's) which makes the block valid.
   - Compute the hash of the block.
   - Use Proof of Work (PoW) to mine the valid block.

   [**Hint:** The transactions are added to depict an exampler scenario and you can add more than 3 if you want to. All the transaction should be signed and you should follow all the tenants of the transaction details (like signature checking)]

3. **Mining Tasks:** Perform an anlaysis of mining difficulty and processing time. **(10 marks)**
   - After mining the blocks, the next task is to analyse the impact of an increase in difficulty (*hint: number of leading zeros in target hash*) on the processing time when executed on your personal machine. (Start with difficulty as "0" leading zeros and increase it till atleast "10" leading zeros)
   - Find the average number of brute force attempts required for finding the nonce of a given length of difficulty (eg., Let's say the difficulty is "000", then use the same difficulty and analyze your code performance for the different attempts (A graphical (visual) representation of the fetched data is also required).

- Similar to the above task, what is the average time for mining a given block with the set difficulty on your machine? You have to state your machines computational capabilities along with the answer.
- With a given maximum nonce value of 100000, what is the maximum difficulty number that your code could solve? Does the same changes over repetitive executions?

  [**Hint:** It may be possible that you machine is not able to reach till 10 leading zeros, in such a case, you should provide the reasons to justify the same support through the processing time depicted in the plot]

Prerequisites: You may need to setup the environment with the help of the components enlisted below.

- Get the coinables/buidljs libraries into your system from https://github.com/coinables/buidljs (Alternatively you can find it in assignment resources in Ultra).
- For the generation of the QRcodes you can get the library from https://github.com/davidshimjs/qrcodejs/blob/master/qrcode.js (Alternatively you can find it in assignment resources in Ultra).
- For Python 3, e.g., pycharm community addition can be downloaded from https://www.jetbrains.com/pycharm/download (You may use python 3 modules hashlib, ecdsa, json and time). You can use additional libraries if required (e.g., for hex conversion).

You must submit the following:

- One piece of code as a **task1.py** file, clearly differentiating the three sub-tasks. Please make your code clean and clear and easy to follow, with comments as necessary.
- A report that will cover the following points **(do not exceed 500 words)**.
  - information asked in all the three sub-tasks (like random IDs, digital address, hash ID, blockc and transaction details, difficulty and nonce).
  - a discussion on the impact of difficulty on the processing time according to analysis in sub-task 3, a visual plot is also required.
  - provide your views on how challenging it was to mine on your personal computer.

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

Task 2. Interacting with Bitcoin Blockchain and Scripting Language (20 Marks)

In this part of the assignment, you are asked to analyse the basic bitcoin blockchain and its scripting language. You will provide a **script and report** for the following tasks.

1. Using blockchain explorer examine the blocks in the bitcoin blockchain and find the block number starting with 93 (e.g., 93188) and perform the following tasks. **[5 marks, i.e., 1 mark each for the below points]**
   - List the number of confirmations received for this block.
   - List atleast two transactions in this block and provide a link to these transactions. In some cases you may find only one transaction in the block, if so just provide only one and suggest the reason for the same.
   - What was the difficulty for mining for this block and also mention the difficulty on the date you accessed this block. Provide your view on this.
   - What was the reward earned by the miner and what would have been value of the reward (in bitcoins and in GBP) if the reward was paid the day you accessed this block (mention the date also).

- Explore the address where the reward was sent and provide details of first and the recent transactions linked to this address. Choose any one of these transactions and provide the fee paid for this transaction.
  (Hint: The blocks can be checked at https://blockexplorer.com)

2. John has to travel to attend a conference and he has some concerns regarding the devices containing his private keys. He would like to store his bitcoins in such a way that they can not be redeemed without the knowledge of secret code (HINT: a password). Now, lets assume that John stored his bitcoins at the following ScriptPubKey address.

```
OP_SHA256
<987dcca6ea151951c963ce256e3a035b044ee0c597836759e30ca11d08bf74ed>
OP_EQUALVERIFY
```

- Your task is to write a ScriptSig script that can be used to redeem this transaction [HINT: module name]. **[5 Marks]**

You have to provide your thoughts and justification on the following points. **(2 marks)**

- Is password a secure method to protect Bitcoins?
- If you identify any security issue, then will Pay-to-script-hash (P2SH) fix the security issue(s) you identified?

The next task is create a P2SHScript using MultiSig to ensure that Johns bitcoins are spent in a secure manner. This should be a real working script including signature and related keys [HINT: you have to use MultiSig where you can use a green address also but the use of green address is not a necessity]. **[8 Marks]**

A list of script op codes is available at https://en.bitcoin.it/wiki/Script

You must submit the following:

- You must submit the P2SHScript using MultiSig in a text file **(task2.txt)**. You can verify the script using online resources (like, https://siminchen.github.io/bitcoinIDE/build/editor.html). The explanation and other details related to P2SH can be added in the report and not in this **.txt** file.
- A report that will cover the following points **(do not exceed 500 words)**.
  - the answer to various questions in sub-task 1 (blockchain explorer) including the justifications if necessary.
  - the ScriptSig script required in sub-task 2.

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

Task 3. Ethereum Smart Contracts (30 Marks)

In this part of the assignment, you are asked to design and create an Ethereum contract in solidity for the given scenario. You should deploy it on the Ropsten test network and call each of the functions to verify that the system works. You must submit your code as a solidity file **task3.sol**. I will attempt to compile your code at http://remix.ethereum.org/ so make sure it works!.

- **Environment Setup [10 marks]:**
  a. Setup your machine to use ethereum test network (Ropsten) by using metmask. Explain the process and significance of each step followed to setup you machine **[4 marks]**.
     (Hint: You can use metmask extension on google chrome to create metamask wallet on your browser)
  b. You have to configure your system to use Ropsten network and do the following task.
     - Get some test Ethers and add them to your wallet **[6 marks].**

- **E-assignment Scenario for smart contracts [20 marks]:** A teacher wants to use a new online assignement submission system to test the responsiveness of his students for a given assignment. So the plan includes the use of tokens for deployment over ethereum network so that the teachers sets the maximum obtainable marks for a particular assignment for each student. (Hint: for *n* different students, *n* different contarcts will be deployed as marks for all students are to be tracked) So, to do so, design the smart contracts that can accomplish the following tasks.
    a. Teacher allocates the assignement as a token to the students deploying a contract by setting a starting marks for the token (e.g., 100 marks). The assignment is due in 7 days, if they submit by the end of 7th day then full marks will be provided. An additional grace period of 3 days is provided to the students. If a student submits the assignment after the 7th day, the marks will be decreased by 20% per day (on day 8 and day 9). The marks will be halved on day 10 (1.e., will be 50% for 10th day) and zero after the deadline is over **[10 marks]**.
    b. You have to implement this smart contract, compile and deploy it on Ganache (https://www.trufflesuite.com/ganache) and remix IDE (http://remix.ethereum.org/). You have to provide the cost of the smart contracts in terms of gas consumption and ether (wei) for both Ganache and remix IDE **[8marks]**. Additional details for comparision will attaract additional marks **[2marks]**.

You must submit the following:

- One piece of code as a **task3.sol** file. Please make your code clean and clear and easy to follow, with comments as necessary.
- A report that will cover a brief discussion on the experimental setup and the smart contract (including the snapshots) **(do not exceed 500 words)**.
    - a comparison on Ganache and Ropsten outcomes.

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

Task 4. Cryptocurrency of the future (15 marks, 500 words max)

Provide you choice on the cryptocurreny of the future.

- You have to provide your assessment of the selected currency, keeing in view of the significant improvement it has witnessed over the past years. Include in your report the justifications on your stance of your chosen currency covering:
    a. the technical differences between your chosen currency and bitcoin;
    b. the idea behind the currency (why was it developed?);
    c. what is performance till now;
    d. how it is mined, and the level of activity of miners;
    e. the impact of mining on energy; and
    f. any notable issues (security concerns) in history of the currency.

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

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 (your should submit a single report only).
- Don't use figures or snapshots merely as a way of circumventing the word limit.
