University of London International Programmes

Computing and Information Systems/Creative Computing

CO3326 Computer security

Coursework assignment 2 2017--18

IMPORTANT: you have been allocated a unique block from the Bitcoin blockchain (with small adjustments), which is to be used for the exercise in this coursework assignment. You can obtain your block using your Student Reference Number (SRN) from the following URL: http://foley.gold.ac.uk/cw18/api/cw2/{srn}. For example, if your SRN is 077242419, you would obtain your block from: http://foley.gold.ac.uk/cw18/api/cw2/077242419.

If you have difficulties obtaining your block, please email us at intcomp@gold.ac.uk.

Background

You may have noticed that recently there has been a considerable amount of attention, investment - and frankly hype - directed towards cryptocurrencies, blockchain and Bitcoin. This coursework assignment is designed to help you extend your knowledge in this area by encouraging self-study and creativity. The web has plenty of information on the subject, so you will not find it difficult to research. Your reading should cover the following topics:

- The general idea behind blockchain.
- The inner workings of Bitcoin.
- Cryptographic hash functions and security of SHA-256.
- Merkle trees.
- Proof-of-work algorithms.

Here are a few starting points for your research:

- The original Bitcoin paper.
- Chapter 6 (pages 63--68) of the subject guide, including the suggested readings.
- Hashcash.

The coursework is composed of two parts, a **report**, which counts as **60%** of your coursework assignment mark, and an **exercise**, which counts as **40%** of your mark. In the *report* you should answer the questions laid out below. In the *exercise* you should validate (or *mine*) the Bitcoin block that you have been provided with, and submit the result in a specific format.

To solve the exercise, you may find it necessary to write a program. You are welcome to use any programming language and you are welcome to use any 3rd party libraries available for SHA-256 and JSON. Libraries are available for most languages, including - and not limited to - Java, C/C++, Scala, Python, JavaScript). Please include key snippets of your code as an annex to your report.

You should read the coursework assignment carefully and pay particular attention to the Submission requirements.

Report: Questions

Please answer the questions briefly, in your own words and to the point. Use diagrams where possible and explain them. Copy-pasting Wikipedia articles or verbose explanations will not get you far.

Question 1

Explain briefly what is meant by *Blockchain* being referred to as:

`"Ledger - Trust + Cryptography"`.

Question 2

In general terms, how does sending money in Bitcoin work? What is the send-to address? Why can't messages be altered by nodes on the network?

Question 3

What are the *inputs* and the *outputs* of a transaction? Instead of a ledger of balances, what do the Bitcoin nodes actually keep track of? What does it take to figure out your own balance?

Question 4

What is *double spending* in Bitcoin?

Question 5

What is the difference between a *transaction chain* and the *block chain*?

Question 6

Describe briefly how Merkle Trees help in pinpointing a mismatching transaction between two blocks without comparing each and every transaction.

Question 7

How does Bitcoin deal with creating a consensus between all nodes on the network with regards to the ordering of transactions?

Question 8

Explain briefly what it would take for Alice to defraud Bob and how Bitcoin prevents this.

Question 9

What would happen if 90% of Bitcoin miners were suddenly shut out of the network for a prolonged period?

Reminder

Don't forget to acknowledge all sources. Make sure you acknowledge any code re-use. It is important that your submitted coursework assignment is your own individual work and, for the most part, written in your own words. You must provide appropriate in-text citation for both paraphrase and quotation, with a detailed reference section at the end of your coursework. Copying, plagiarism and unaccredited and wholesale reproduction of material from books or from any online source is unacceptable, and will be penalised (see our guide on how to avoid plagiarism on the VLE).

Exercise

You have been provided with an *un-mined block* in the following format (this is an example for illustration):

```
{
  "srn": "077242419",
  "name": "Carl Davis",
  "block": {
    "bits": "1d00ffff",
    "confirmations": 505791,
    "difficulty": 3,
    "hash": "0f230ad3fc6424759d5f079e6c5d96f60b9821b998a576d34ab6844ef82aba7b",
    "height": 102,
    "isMainChain": true,
    "merkleroot":
"3b44b5ce02d36db604f0d6b7cc761685484c370a235c54539700b1ad23afefce",
    "nonce": 0,
    "previousblockhash":
"00000000b69bd8e4dc60580117617a466d5c76ada85fb7b87e9baea01f9d9984",
    "reward": 50,
    "size": 215,
    "time": 1231662670,
    "tx": [
      "3b44b5ce02d36db604f0d6b7cc761685484c370a235c54539700b1ad23afefce"
    ],
    "version": 1
  }
}
```

For this *block*, Carl Davis would submit the following JSON, which reflects a correct solution:

```
{
  "srn": "077242419",
  "name": "Carl Davis",
  "block": {
    "bits": "1d00ffff",
    "confirmations": 505791,
    "difficulty": 3,
    "hash": "000fe96859f2134582a253edfca6035202874168d28e9cf923e2d1295f9cc97c",
    "height": 102,
    "isMainChain": true,
    "merkleroot":
"3b44b5ce02d36db604f0d6b7cc761685484c370a235c54539700b1ad23afefce",
    "nonce": 5684,
    "previousblockhash":
"00000000b69bd8e4dc60580117617a466d5c76ada85fb7b87e9baea01f9d9984",
    "reward": 50,
    "size": 215,
    "time": 1231662670,
    "tx": [
      "3b44b5ce02d36db604f0d6b7cc761685484c370a235c54539700b1ad23afefce"
    ],
    "version": 1
  }
}
```

Explanation

These JSONs have been indented here for the purpose of explanation. The JSON you have been provided with is in a packed form and you should also submit it in a packed form. So, Carl Davis would actually have received this JSON.

From your readings on *proof-of-work* and *Hashcash* you would realize why this block is invalid (in Bitcoin terms) and you would also know what to adjust to make this a valid block in the block chain. A few clarifications are necessary here:

- SHA-256 is used and it is applied once.
- The hash is computed on the *block* in its packed JSON form. Make sure you remove the `hash` key/value before computing the hash. For example, Carl Davis would compute the hash on the block below (in its packed form):

```
  {
    "bits": "1d00ffff",
    "confirmations": 505791,
    "difficulty": 3,
    "height": 102,
    "isMainChain": true,
    "merkleroot":
"3b44b5ce02d36db604f0d6b7cc761685484c370a235c54539700b1ad23afefce",
    "nonce": 0,
    "previousblockhash":
"00000000b69bd8e4dc60580117617a466d5c76ada85fb7b87e9baea01f9d9984",
    "reward": 50,
    "size": 215,
    "time": 1231662670,
    "tx": [
      "3b44b5ce02d36db604f0d6b7cc761685484c370a235c54539700b1ad23afefce"
    ],
    "version": 1
  }
```

- Double-check that the block you've been provided with is consistent, *i.e.* the `hash` value corresponds to the actual hash of the block. There are online tools you can use to compute SHA-256, such as this. For example, Carl Davis would double-check that the hash of the following JSON string is `0f230ad3fc6424759d5f079e6c5d96f60b9821b998a576d34ab6844ef82aba7b`:

```
{"bits":"1d00ffff","confirmations":505791,"difficulty":3,"height":102,"isMainChain"
:true,"merkleroot":"3b44b5ce02d36db604f0d6b7cc761685484c370a235c54539700b1ad23afefc
e","nonce":0,"previousblockhash":"00000000b69bd8e4dc60580117617a466d5c76ada85fb7b87
e9baea01f9d9984","reward":50,"size":215,"time":1231662670,"tx":["3b44b5ce02d36db604
f0d6b7cc761685484c370a235c54539700b1ad23afefce"],"version":1}
```

- The *difficulty* indicates the number of expected leading zeros in the hash to make it a *valid* block (in Bitcoin terms). Based on your readings you will realize that there is a deviation between this definition of difficulty and the one used in Bitcoin.
- Finally, double-check that the result of your computation is correct. For example, Carl Davis would double-check that the hash of his *mined* block (below) is `000fe96859f2134582a253edfca6035202874168d28e9cf923e2d1295f9cc97c`:

```
{"bits":"1d00ffff","confirmations":505791,"difficulty":3,"height":102,"isMainChain"
:true,"merkleroot":"3b44b5ce02d36db604f0d6b7cc761685484c370a235c54539700b1ad23afefc
e","nonce":5684,"previousblockhash":"00000000b69bd8e4dc60580117617a466d5c76ada85fb7
b87e9baea01f9d9984","reward":50,"size":215,"time":1231662670,"tx":["3b44b5ce02d36db
604f0d6b7cc761685484c370a235c54539700b1ad23afefce"],"version":1}
```

Submission requirements

You should upload **two** single files only. These must not be placed in a folder, zipped, *etc.*

The **report** should be submitted as a PDF document following a *strict naming scheme*: `YourName_{srn}_CO3326_cw2.pdf`. For example, Carl Davis with SRN 077242419 would submit `CarlDavis_077242419_CO3326_cw2.pdf`.

The **exercise** should be submitted as a JSON file with a *strict format* and *strict naming scheme*. The exercise will be automatically checked by an algorithm, so pay particular attention to its format. The name of the file should be `YourName_{srn}_CO3326_cw2.json`; for example, Carl Davis with SRN 077242419 would submit `CarlDavis_077242419_CO3326_cw2.json`.

NOTE: As the JSON is evaluated by an algorithm, every quote, comma, colon, curly brace upper/lower case is crucial. Please pay attention to these. It would be a shame to lose a potential

40% of the total marks for this coursework assignment because of a misplaced comma or a missing quote. There are online tools you can use for JSON formatting and validation (for example this), so double-check that your JSON is syntactically correct.