

# Blockchain and Cryptocurrency Summative

wcrr51

2022

## Task 1: Blockchain Design and Implementation

### Random IDs

The HTML/CSS/JavaScript web page shown in Figure 1 generates a given number of address and private key pairs for use in transactions, also outputting the addresses as QR codes for ease of use.

**Finance DApp Credential Generator**

Credential Count




Party ID	Credentials	QR Code
A	Address 1DwWm1eVwwWTijVGN9kNAWCQWitmTFLEt4 Private Key L2kG6xdPHH3uE5Nes4P3TfX6KzCn982gdPfhTncaZcfKuGcFgFYL	
B	Address 15zxhoEsJw2xBMPUxVrBpepR36VZhUvzek Private Key L1VMTbXn8CBvbuJNHMTEX4qYAb7mFUtQeqVpCrDUkMCXfUmV1Vsh	
C	Address 1DnNkHkmYZrJzmqSRVb967ZZs4un2RwTet Private Key KwkykaU9evqi37qP8aEDimJM5zxDuGVrbr9E5LHkb9o4EZjMkT24	

Figure 1: Screenshot of the credential generation web page with 3 freshly generated address and private key pairs.

The the sample data generated on the web page is repeated below and will be used for demonstration transactions in the python blockchain implementation.

Party A:

Address 1DwWm1eVwwWTijVGN9kNAWCQWitmTFLEt4

Private Key L2kG6xdPHH3uE5Nes4P3TfX6KzCn982gdPfhTncaZcfKuGcFgFYL

Party B

Address 15zxhoEsJw2xBMPUxVrBpepR36VZhUvzek

Private Key L1VMTbXn8CBvbuJNHMTEX4qYAb7mFUtQeqVpCrDUkMCXfUmV1Vsh

Party C:

Address 1DnNkHkmYZrJzmqSRVb967ZZs4un2RwTet  
Private Key KwkykaU9evqi37qP8aEDimJM5zxDuGVrbr9E5LHkb9o4EZjMkT24

## Implementation Information

The python blockchain implementation provides classes for Transaction, Block, and Blockchain code representations, each one can be serialised and deserialised to and from binary (as `bytes`), or serialised to JSON (for user friendliness and readability). All hashes are calculated using the binary-serialised representations for better speeds and data consistency. Each transaction allows for a variable number of account inputs and outputs, and inputs need to be signed using a private key. The difficulty value for each block is the number of leading zeros in the hash it has mined to (through brute force changing of the nonce value). Figures 2 and 3 show the two blocks, and Figure 4 shows the validation of the blockchain.

```
Creating and mining block... Done!
Genesis Block:
{
  "VERSION": 1,
  "difficulty": 5,
  "hash": "000001cdad528e0694a48f4959a945e22630d1cd2bb46689a24d1f8661bd9e58",
  "hash_previous_block": "0000000000000000000000000000000000000000000000000000000000000000",
  "id": 0,
  "nonce": 33530,
  "timestamp": "2022-01-15T14:30:00",
  "transactions": [
    {
      "VERSION": 1,
      "hash": "cc66dba245b86b955a84e16fd393c77afc83c958a359ef3f4e3217250ca034bf",
      "inputs": [],
      "outputs": [
        {
          "3f51e1091f1b33a7d3767ec0ac2e7e283cebdf5c53c638ad54d69ad41a1e45df13c97f836e6494e050297ed457e0912b5e6f6d89e2281fb7fe2759463aa33596": 5000
        }
      ]
    },
    {
      "VERSION": 1,
      "hash": "cc421acfb1a53a517a62d339a62c8c228c8b7f952561a622491659d01621414e",
      "inputs": [],
      "outputs": [
        {
          "153633c1f6c56388ef35b3e4b34fdb00409e673e5ab9fc742710e2a9251bfec019db1ec539f6460455dc6d7e99d9b803a7f334881a6b3454eb8630f2db45031b": 10000
        }
      ]
    },
    {
      "VERSION": 1,
      "hash": "8d0509b07f8e8a6ce2e4d3650cedcb4c43aff3da114884b1f26c7e87b3880039",
      "inputs": [],
      "outputs": [
        {
          "38b93f9ebdfc9d1c326f14bfc62d3d2a98571d48c21c38914d1d4db3dfecfb576267437eed5428a6e2fb82f588413f69e5840d90dd29e14743a12a92aab57b": 2000
        }
      ]
    }
  ]
}
```

Figure 2: Console output of the first block.

```
Creating and mining block... Done!
Block #1:
{
  "VERSION": 1,
  "difficulty": 5,
  "hash": "000005504e0195396b0387233d2416162bab7c7879549c6f0af7d41e218dbd5",
  "hash_previous_block": "000001cdad528e0694a48f4959a945e22630d1cd2bb46689a24d1f8661bd9e58",
  "id": 1,
  "nonce": 197192,
  "timestamp": "2022-01-15T14:30:00",
  "transactions": [
    {
      "VERSION": 1,
      "hash": "346fec35d89d0bab5ee650cb259f9d9ad230ca60887bbd2464idd26f823f592",
      "inputs": [
        {
          "public_key": "3f51e1091f1b33a7d3767ec0ac2e7e283cebdf5c53c638ad54d69ad41a1e45df13c97f836e6494e050297ed457e0912b5e6f6d89e2281fb7fe2759463aa33596",
          "signature": "1a0a80d368b96034613cce997ec6597921c98fca6e96e456da8504d95b6c01db142c36f22072bb010afc204e93bc22255959a901f862f1460e0002c03c56a326"
        }
      ],
      "outputs": [
        {
          "153633c1f6c56388ef35b3e4b34fdb00409e673e5ab9fc742710e2a9251bfec019db1ec539f6460455dc6d7e99d9b803a7f334881a6b3454eb8630f2db45031b": 1000,
          "38b93f9ebdfc9d1c326f14bfc62d3d2a98571d48c21c38914d1d4db3dfecfb576267437eed5428a6e2fb82f588413f69e5840d90dd29e14743a12a92aab57b": 2000
        }
      ]
    },
    {
      "VERSION": 1,
      "hash": "4c3f20d1b9e6d9154ccfc6e39d6db9110f399a5e38837337e51f640028ae2alc",
      "inputs": [
        {
          "public_key": "153633c1f6c56388ef35b3e4b34fdb00409e673e5ab9fc742710e2a9251bfec019db1ec539f6460455dc6d7e99d9b803a7f334881a6b3454eb8630f2db45031b",
          "signature": "eaf2f08a6e80196f48d53e2844199b3fd96169fe2bc617e75eb692ff7c55c66622810e95f8c4c7ba53ab207dcef3f015ff15525ef4e6de854884cbeaaf0af43a"
        }
      ],
      "outputs": [
        {
          "38b93f9ebdfc9d1c326f14bfc62d3d2a98571d48c21c38914d1d4db3dfecfb576267437eed5428a6e2fb82f588413f69e5840d90dd29e14743a12a92aab57b": 2000
        }
      ]
    },
    {
      "VERSION": 1,
      "hash": "776b2808caf7422b177956b27464ff1f107cd5b4e6059ca5d328cb9f17ddd417",
      "inputs": [
        {
          "public_key": "38b93f9ebdfc9d1c326f14bfc62d3d2a98571d48c21c38914d1d4db3dfecfb576267437eed5428a6e2fb82f588413f69e5840d90dd29e14743a12a92aab57b",
          "signature": "f19e3ba83af8a974894b843e592fa38a03d312c1f50baa211c44b513cfa88d08e949d04ccae69b5eabafa27da350fe2cfab83b3edf7cd913243e4a8f4ecf4"
        }
      ],
      "outputs": [
        {
          "153633c1f6c56388ef35b3e4b34fdb00409e673e5ab9fc742710e2a9251bfec019db1ec539f6460455dc6d7e99d9b803a7f334881a6b3454eb8630f2db45031b": 10
        }
      ]
    }
  ]
}
```

Figure 3: Console output of the second block.

```

Verifying blockchain with 2 blocks...

- Checking validity of genesis block...
  ✓ The hash of block 0 matches the stored hash.
  ✓ Block 0 has sufficient proof of work.
  ✓ The Merkle root hash of block 0 matches the stored Merkle root hash.
  - Checking validity of transaction 'cc66dba245b86b955a84e16fd393c77afc83c958a359ef3f4e3217250ca034bf'
    ✓ Number of available signatures matches number of inputs.
    ✓ Successfully validated transaction.
  - Checking validity of transaction 'cc421acfb1a53a517a62d339a62c8c228c8b7f952561a622491659d01621414e'
    ✓ Number of available signatures matches number of inputs.
    ✓ Successfully validated transaction.
  - Checking validity of transaction '8d0509b07f8e8a6ce2e4d3650cedcb4c43aff3dall4884b1f26c7e87b3880039'
    ✓ Number of available signatures matches number of inputs.
    ✓ Successfully validated transaction.
  ✓ All transactions in block 0 successfully validated.
  ✓ Genesis block passed validation.

- Checking validity of block 1...
  ✓ Block 1's previous hash matches that of the previous block.
  ✓ The hash of block 1 matches the stored hash.
  ✓ Block 1 has sufficient proof of work.
  ✓ The Merkle root hash of block 1 matches the stored Merkle root hash.
  - Checking validity of transaction 'dff439576b553b043614c44a027d11923602256fac7f3aa0664f285be3d265e2'
    ✓ Number of available signatures matches number of inputs.
    ✓ Signature authenticated for address beginning '3f51e1091f1b33a7d3767ec0ac2e7e28'.
    ✓ Successfully validated transaction.
  - Checking validity of transaction 'cf8973c7336d9200d472f1e16600603204931bald2b3976373290b081486a27a'
    ✓ Number of available signatures matches number of inputs.
    ✓ Signature authenticated for address beginning '153633c1f6c56388ef35b3e4b34fdb00'.
    ✓ Successfully validated transaction.
  - Checking validity of transaction 'c0761906a23949381d63599b31c872c8f3a32f6b31466d646681d5253e4eab72'
    ✓ Number of available signatures matches number of inputs.
    ✓ Signature authenticated for address beginning '38b93f9ebdfc9d1c326f14bfc62d3d2a'.
    ✓ Successfully validated transaction.
  ✓ All transactions in block 1 successfully validated.
  ✓ Block 1 passed validation.

Blockchain successfully validated!

```

Figure 4: Console output of the blockchain validity check (with `verbose = True`).

## Mining Analysis

In the experimental setup, mining was performed on the above two blocks for difficulties in the range of  $[0, 10]$ , measuring the average nonce value, average time, and total time across both blocks. These results are summarised linearly in Figures 5, 6, and 7. Across both blocks, the total mining time, mean running time, and mean nonce values all clearly show an exponential increase from an increment of difficulty. Hence, this data is also demonstrated on logarithmic graphs in Figures 8, 9, and 10.

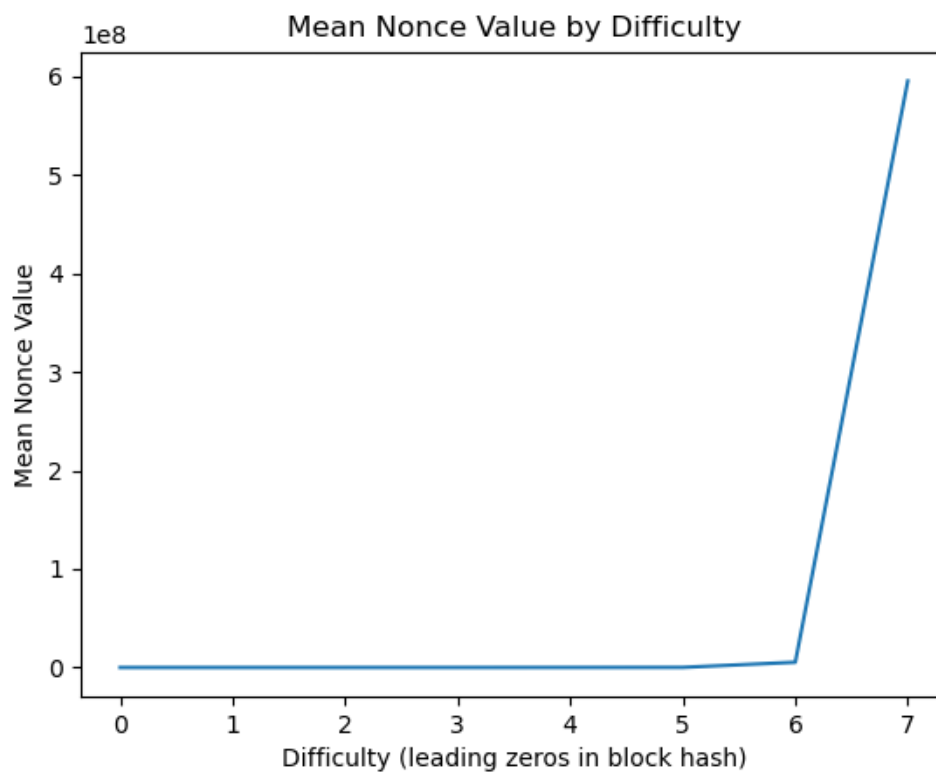


Figure 5: Mean nonce value by difficulty on a linear scale.

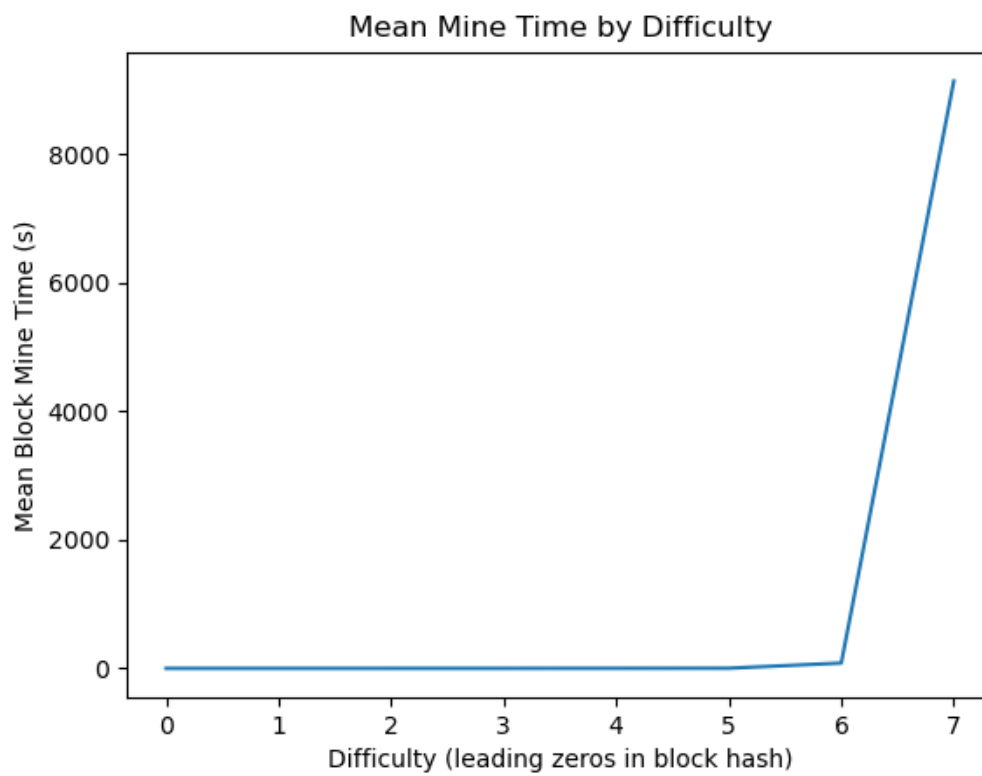


Figure 6: Mean time taken by difficulty on a linear scale.

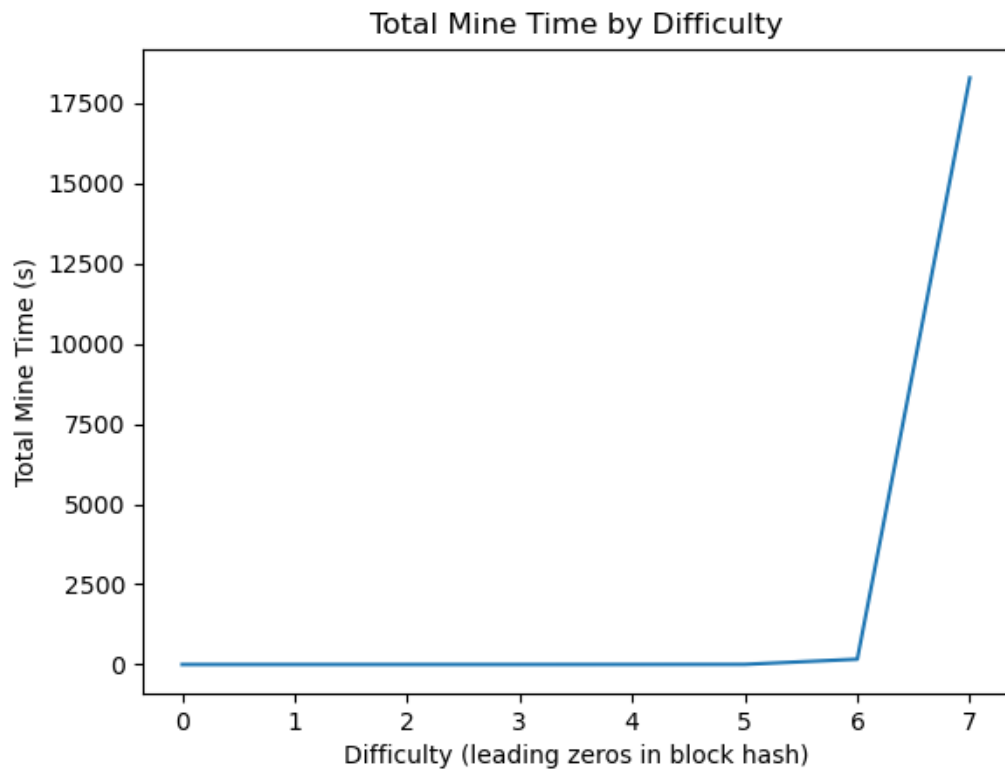


Figure 7: Total time taken by difficulty on a linear scale (across two blocks).

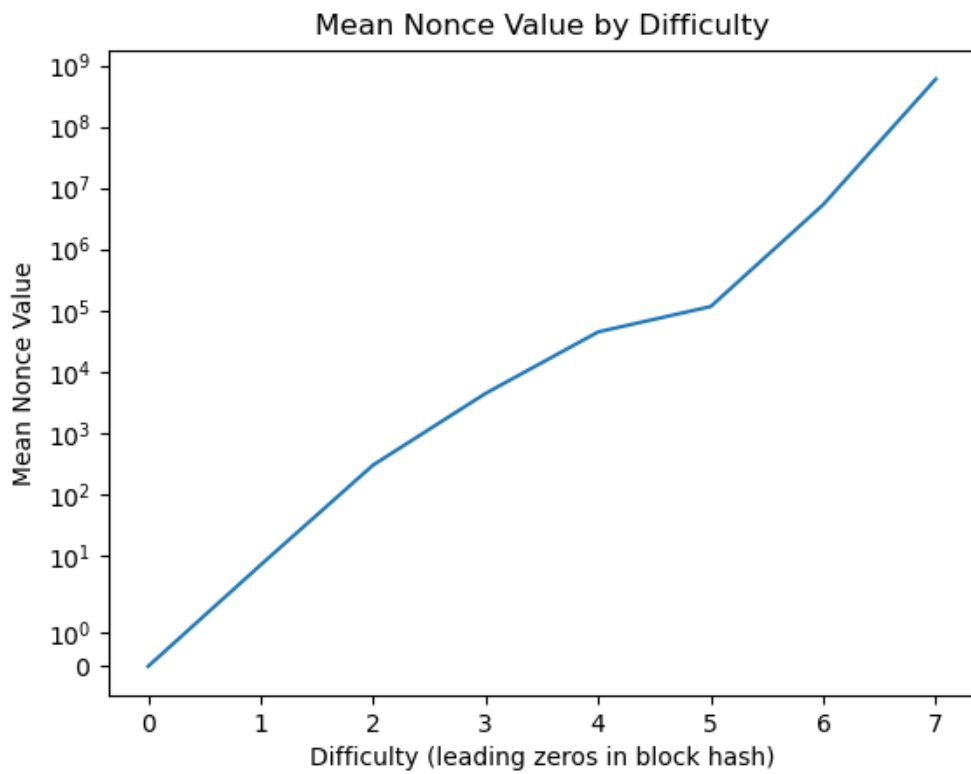


Figure 8: Mean nonce value by difficulty on a log scale.

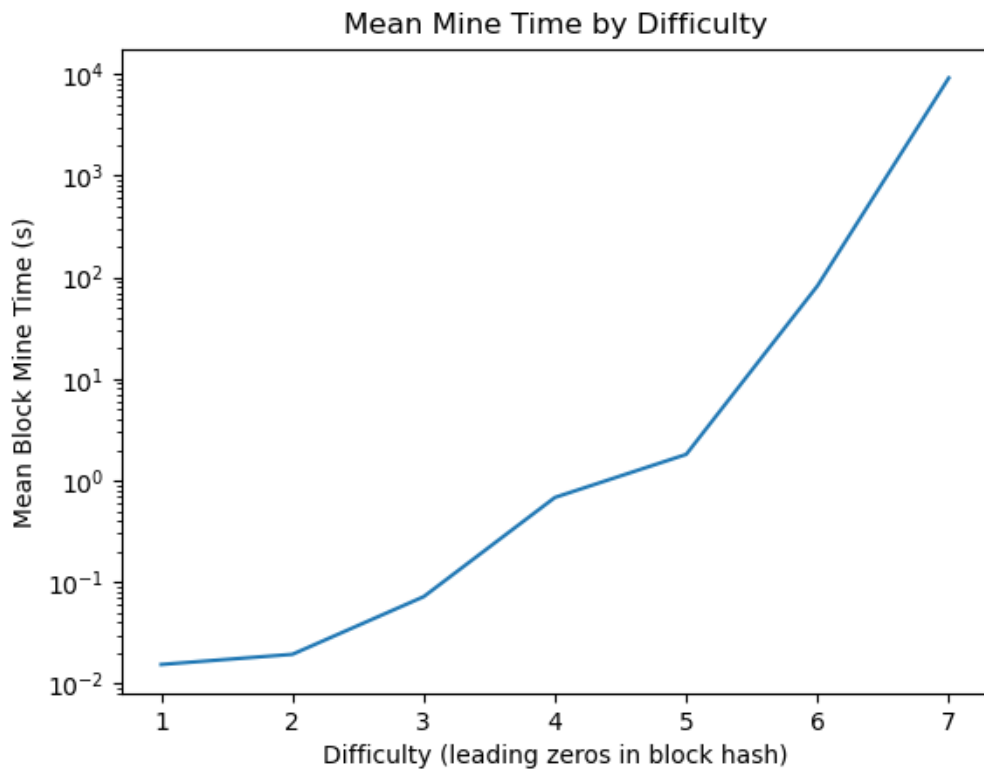


Figure 9: Mean time taken by difficulty on a log scale.

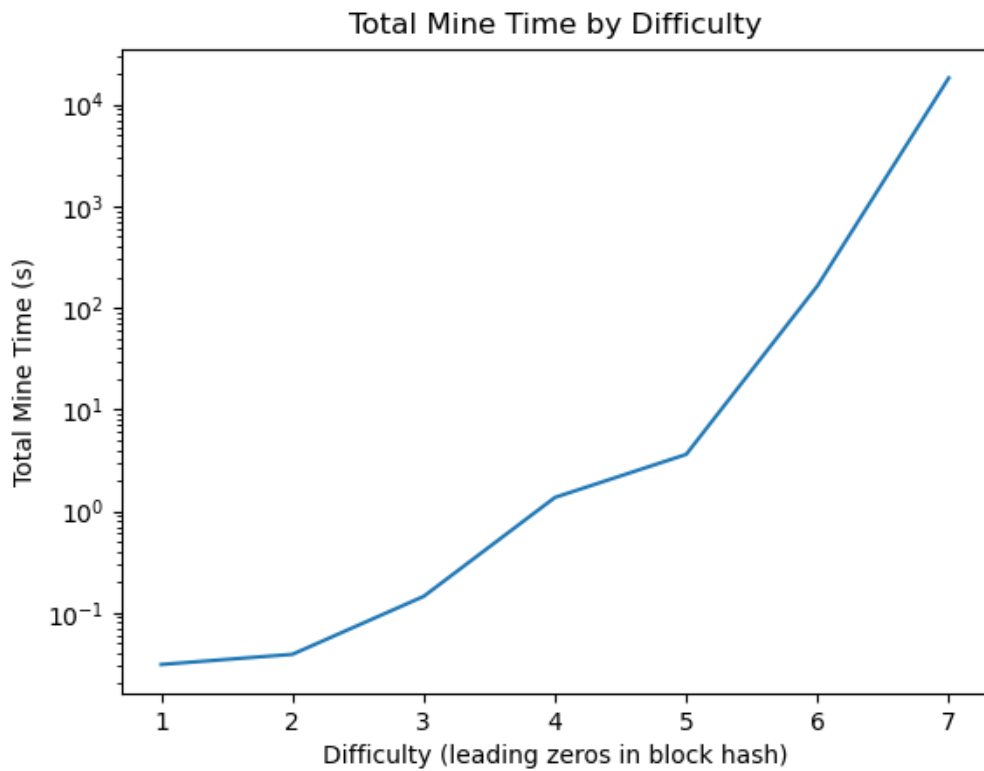


Figure 10: Total time taken by difficulty on a log scale (across two blocks).

The console outputs of the mining analysis are shown in Figure [fig:log-4]. The actual nonce values in the figures and logs represent the number of brute force attempts required to crack a given level of difficulty.

Further to this, Figure [fig:log-4] clearly demonstrates that the maximum difficulty attainable for a maximum nonce value of 100,000 is 5 for block 0, and 4 for block 1. This also demonstrates that the minable difficulty is different under different executions. If true timestamping was enabled (instead of fixed), there would be even more variation in this value,

```
Blockchain successfully validated!
Analysing mining difficulty 0...
Block 0 mining time: 0.0140 s, nonce: 0
Block 1 mining time: 0.0160 s, nonce: 0
Overall time taken: 0.0300 s
Average block mine time: 0.0150 s
Average nonce: 0

Analysing mining difficulty 1...
Block 0 mining time: 0.0150 s, nonce: 2
Block 1 mining time: 0.0160 s, nonce: 13
Overall time taken: 0.0310 s
Average block mine time: 0.0155 s
Average nonce: 7

Analysing mining difficulty 2...
Block 0 mining time: 0.0150 s, nonce: 45
Block 1 mining time: 0.0240 s, nonce: 553
Overall time taken: 0.0390 s
Average block mine time: 0.0195 s
Average nonce: 299

Analysing mining difficulty 3...
Block 0 mining time: 0.1030 s, nonce: 6438
Block 1 mining time: 0.0410 s, nonce: 2403
Overall time taken: 0.1450 s
Average block mine time: 0.0720 s
Average nonce: 4420

Analysing mining difficulty 4...
Block 0 mining time: 1.0360 s, nonce: 68785
Block 1 mining time: 0.3280 s, nonce: 20108
Overall time taken: 1.3640 s
Average block mine time: 0.6820 s
Average nonce: 44446

Analysing mining difficulty 5...
Block 0 mining time: 0.4990 s, nonce: 33530
Block 1 mining time: 3.1170 s, nonce: 197192
Overall time taken: 3.6160 s
Average block mine time: 1.8080 s
Average nonce: 115361

Analysing mining difficulty 6...
Block 0 mining time: 122.8970 s, nonce: 7966481
Block 1 mining time: 41.9937 s, nonce: 2674474
Overall time taken: 164.8907 s
Average block mine time: 82.4454 s
Average nonce: 5320477

Analysing mining difficulty 7...
Block 0 mining time: 15325.4148 s, nonce: 1003901512
Block 1 mining time: 2958.3905 s, nonce: 186968261
Overall time taken: 18283.8074 s
Average block mine time: 9141.9027 s
Average nonce: 595434886
```

Figure 11: Console output of the mining analysis.

The implemented code was run on a windows system with an i7 6700k processor and 16GB DDR4 memory. Beyond a difficulty of 6 leading zeros, the computer began freezing (this may be due to its overclocking instability), and after leaving it mining a difficulty of 8 for many hours, no results were found (this is in line with the exponential nature of increased difficulty).

In fact, after extended time minding a difficulty of 8, the program crashed due to the size of the nonce exceeding the 4-byte allocation, this is a clear issue of short-sightedness with the implemented blockchain that needs to be fixed (for example by upping the serialised nonce value to 8 bytes).

In spite of this, its still likely that, without parallelisation, the time taken for the mining to complete for a difficulty of 8

would go beyond 10 hours of processing time. Hence, with the current mining algorithm, it is clear from the figures that a time in the order of days and a nonce value of  $10^{12}$  would be required to reach difficulty 8.



## Task 2: Interacting with Bitcoin Blockchain

### Sub-task 1

Analysing Block 93000 (<https://www.blockchain.com/btc/block/93000>), mined November 20th 2010 at 16:41 GMT by the account of address 1ECxGytJ7sNV1StDVmCmTe662rXDjZsMGg.

As of 14/02/2022:

- It has 631,354 confirmations.
- 4 transactions total, three of which are listed below.
- The mining difficulty was 6,866.90 ( $6.8 \times 10^3$ ) (the hash of the block (starting 0000000000015d5379547ad4) contains 11 leading zeros). At date of access, the mining difficulty is around  $2 \times 10^{13}$  (about 10 orders of magnitude difference in around 10 years, with around 19 leading zeros in modern block hashes).
- The miner received 50.02 BTC, worth about £9.35 in 2010 (given an exchange rate back then of 1 USD is 0.65 GBP) and £1,400,895.07 today (February 2022, given an exchange rate of 1 USD is 0.74 GBP).
- The address to which the reward was sent (1ECxGytJ7sNV1StDVmCmTe662rXDjZsMGg) has only had a single transaction (the coinbase transaction for this block). The 50.02 BTC from the coinbase transaction has remained in the account since 2010, hence they have never paid any transaction fees. The fees paid for the other transactions in this block were 0.01 BTC. Modern day fees for small transfers are typically around 0.0005 BTC (for example, transaction 4396f4a0168295d2b976ab12b634c8148ba35c5a9ce7ee98c7b1a5a74687420c).

Three of the transactions are as follows:

- A coinbase (coin creation) transaction with no fee, creating and depositing 50.02 BTC into the account of address 1ECxGytJ7sNV1StDVmCmTe662rXDjZsMGg (block miner's address).  
<https://www.blockchain.com/btc/tx/29e4ce006ce48742f5f6c4698b5ad8330fabf8bc1b73e3895c417d5ce7467715>
- A coin transfer from account of address 1JNYdVexau4GMBnEdLnHQuuvdyuaQCaVgf to account of address 1GMGVXYqqQonaHDriiUKtCZLoGGPzRqY6x totalling 453.51 BTC, and to account of address 1FBc7u78vYsLAheeRg4eq4SMoj5QHo92pC totalling 0.5 BTC, paying a 0.01 BTC transaction fee.  
<https://www.blockchain.com/btc/tx/0784b97ab0d6936214634ae65f7b4ca1bcdda1600b6a43cfff3d53e76e228bfd>
- A coin transfer from account of address 1GNFybXEuz7b6uhXHFUna1SkUDrsCsXhJf to account of address 1KcJWvKFJW9F1yGqNvSyrzQ2Te52hm4mn7 totalling 0.5 BTC, and to account of address 17PeTDGkQ8q2MkLu24KGFqjbH1ZjDr7dP2 totalling 87.76 BTC, paying a 0.01 BTC transaction fee.  
<https://www.blockchain.com/btc/tx/140cfff0021c7dee5b4b61dbd2472c3da1ec20ca3564a676881eaae95dd4c23>

### Sub-task 2

The following ScriptSig can be used to redeem John's transaction.

```
<blockchain and cryptocurrencies>
```

This combines with the ScriptPublicKey to create the following valid script.

```
<blockchain and cryptocurrencies>
```

```
op_sha256
```

```
<987dcca6ea151951c963ce256e3a035b044ee0c597836759e30ca11d08bf74ed>
```

```
op_equalverify
```

As this shows, a password is clearly not a secure method to protect Bitcoins because the ScriptSig is stored in the transaction, meaning, once a transaction is made, the password is publicly available. This means it cannot be used again, and, at worst, can be read by others who could potentially publish an adversarial transaction before the legitimate one is accepted.

P2SH would improve but not necessarily fix security, when using green addresses there is potential for an untrustworthy key-pair operator to double spend, thereby voiding the effectiveness of a decentralised blockchain.

## Task 3: Ethereum Smart Contracts

### Environment Setup

Firstly, install the MetaMask Chrome Browser Extension through the Chrome Web Store. Once installed, open the extension and view the setup page. Here, opt to create a new Wallet and make a password for MetaMask.

The next steps are of significance. You are now given a secret phrase, if lost there is no way of accessing your accounts through MetaMask. If it gets leaked, anyone can access your Wallet and accounts. Make sure this is accessible to you but still secure (you will be tested on these phrases and their ordering on the next page).

A default account will be made for you, starting with 0 ETH on the Ethereum Mainnet.

Importantly, to change network you must first show the test networks. To do this, go to the top right and click on your profile, click settings, Advanced, and scroll to and enable the 'Show test networks' setting. Now go to the top right (where it says 'Ethereum Mainnet', click the drop-down, and select 'Ropsten Test Network'.

You are now set up and ready to use MetaMask on the Ropsten Test Network.

To get some test Ethers on Ropsten, head to <https://faucet.metamask.io/> (or a similar trustworthy faucet website), and click 'request 1 ether from faucet', this will prompt you to link your MetaMask account to the MetaMask ether faucet. Once this is done, you can press the button more times to get more Ether (your MetaMask extension should now look like Figure 12).

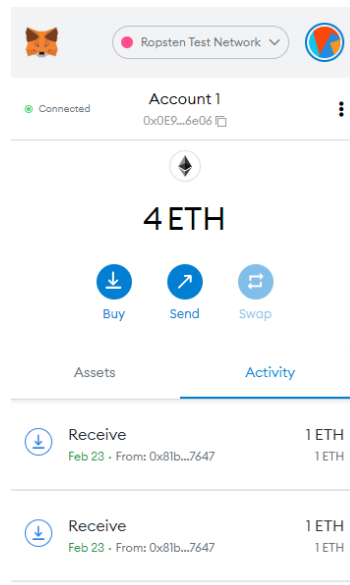


Figure 12: MetaMask wallet on the Ropsten Test Network with 4 ETH from the MetaMask Ether Faucet.

## E-assignment Smart Contracts

### Implementation

The implementation consists of two contracts: one for the assignment, and another for each student undertaking the assignment. The **Assignment** contract keeps track of all students and stores maps from their names and contract hashes to their contracts. The **Student** contract stores the name, maximum marks, and deadline timestamp for each student taking the assignment, the `getCurrentMark()` function returns the mark of the student at the time of block acceptance (without need to update the contract state and thus pay a transaction fee).

### Ganache and Ropsten Comparison

All with a gas limit of 3,000,000, Figures 13 and 14 show the Ether fees incurred from the publishing the **Assignment** smart contract on the Ganache and Ropsten Test networks respectively, Figures 15 and 16 show the Ether fees incurred from publishing one instance of the **Student** smart contract on the Ganache and Ropsten Test networks respectively.

The Ropsten Test network has a significantly higher gas fee than that of the local Ganache network.

A further notable comparison is that, while the publication transactions were instant on the local Ganache network, they took about 10 seconds to be verified on the Ropsten network.

The screenshot shows the Remix IDE interface for the Ganache Local network. At the top, it says 'Ganache Local'. Below that, there's a header with 'Account 3' and a 'New Contract' button. The main area shows the URL 'http://remix.ethereum.org' and a 'CONTRACT DEPLOYMENT' button. Below this, there are tabs for 'DETAILS' and 'DATA'. The 'DETAILS' tab is active, showing the 'Estimated gas fee' as 0.02795624 ETH. It also shows 'Site suggested' and 'Max fee: 0.02795624 ETH'. Below this, the 'Total' is shown as 0.02795624 ETH, with 'Amount + gas fee' and 'Max amount: 0.02795624 ETH'. At the bottom, there are 'Reject' and 'Confirm' buttons.

Field	Value
Estimated gas fee	0.02795624
Site suggested	Max fee: 0.02795624 ETH
Total	0.02795624
Amount + gas fee	Max amount: 0.02795624 ETH

Figure 13: Fees incurred from publishing the `Assignment` smart contract on the Ganache network.

The screenshot shows the Remix IDE interface for the Ropsten Test Network. At the top, it says 'Ropsten Test Network'. Below that, there's a header with 'Account 1' and a 'New Contract' button. The main area shows the URL 'http://remix.ethereum.org' and a 'CONTRACT DEPLOYMENT' button. Below this, there are tabs for 'DETAILS' and 'DATA'. The 'DETAILS' tab is active, showing the 'Estimated gas fee' as 0.3202403 ETH. It also shows 'Site suggested' and 'Max fee: 0.63698257 ETH'. Below this, the 'Total' is shown as 0.3202403 ETH, with 'Amount + gas fee' and 'Max amount: 0.63698257 ETH'. At the bottom, there are 'Reject' and 'Confirm' buttons.

Field	Value
Estimated gas fee	0.3202403
Site suggested	Max fee: 0.63698257 ETH
Total	0.3202403
Amount + gas fee	Max amount: 0.63698257 ETH

Figure 14: Fees incurred from publishing the `Assignment` smart contract on the Ropsten test network.

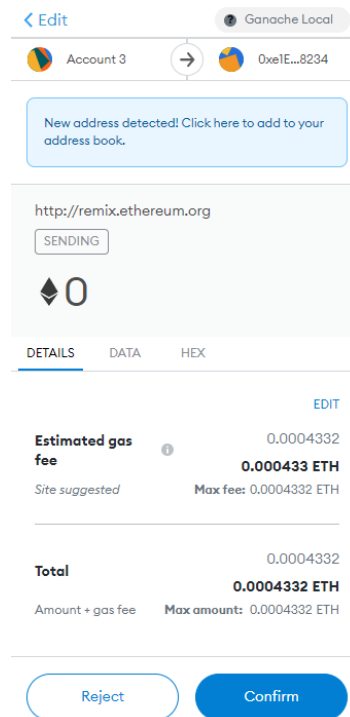


Figure 15: Fees incurred from publishing the **Student** smart contract on the Ganache network.

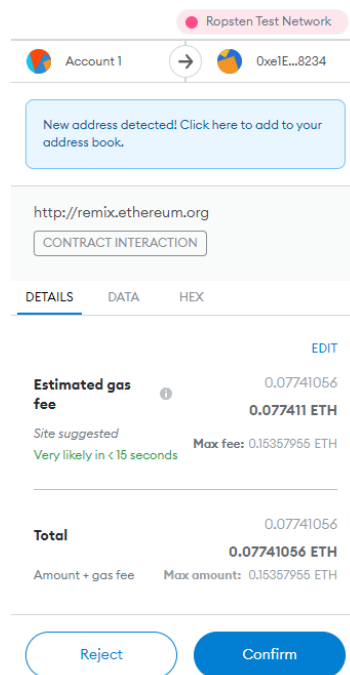


Figure 16: Fees incurred from publishing the **Student** smart contract on the Ropsten test network.

## Task 4: Cryptocurrency of the Future

Cardano [2] (and its native currency ADA) is a third generation, decentralised, public, and open-source blockchain and cryptocurrency launched in 2017 after two years of development. The reason for its development was to take the first two generations and attempt to future proof by building on their advantages and alleviating their downsides, namely scalability, sustainability, and interoperability. This was partially achieved by its development through standalone use of peer-reviewed papers.

To enable transaction transparency, Cardano looks to allow users to seamlessly attach metadata to transactions [1], moving transactions closer to those done more traditionally, while still enabling anonymity and remaining decentralised. This comes with the goals of improving validation and verification, authentication and attribution, providing a secure record of information, and accurate timestamping. In Bitcoin, while it is possible to attach data to transactions, is very limited and lacks official support.

As part of the drive for scalability and lower energy demands, compared to Bitcoin's computationally (and thus energy) intensive Proof-of-Work consensus algorithm, Cardano adopts a Proof-of-Stake (PoS) blockchain, and is currently the largest PoS blockchain/cryptocurrency in widespread use. On a technical level, whereas in PoW every peer is capable of verifying and mining blocks (by solving a computationally-heavy task), in PoS, the network randomly divides block-mining times (epochs) into slots, for which it selects stakeholders (those who already have ADA) as slot-leaders to manage the verification of the blocks in their assigned slot.

One key metric when it comes to scalability is transactions per second (TPS). Under PoS, Cardano is capable of 250 TPS, already a significant improvement compared to Bitcoin's maximum theoretical 27 TPS [7], or between 3.3 and 7 TPS [4] when using median average transaction sizes, due to its relatively long 10 minute block time and lower 1MB per block size limit. However, with the introduction of the Hydra scaling solution [3] at some point in 2022, Cardano will be capable of 1-2 million TPS, going far beyond Visa's 24,000 [6].

Finally, compared to Bitcoin's purely peer-to-peer architecture, Cardano uses a Recursive InterNetwork Architecture (RINA) [5], where peers exist within a sub-network who are capable of working independently, allowing the sub-networks to, for the most part, work independently of each other, decreasing local latency and overall scalability complexity.

As of today, Cardano is sitting at just under 1 GBP per ADA, having peaked in September 2021 at just over 2 GBP per ADA, with no significant relative deviation from Bitcoin over the last year. As [6] notes, there is speculation that the introduction of Hydra will boost Cardano's performance.

From a security standpoint, as part of an attempt to improve its blockchain security, Cardano have a scheme to encourage hackers to report vulnerabilities in return for monetary compensation [1]. Fortunately, their theory-before-practice approach with the hindsight of first and second generation cryptocurrencies has enabled them to avoid any major security concerns in the relatively short history of the currency.

## References

- [1] Justinas Baltrusaitis. *Cardano announces the doubling of rewards for hackers to spot possible vulnerabilities*. Feb. 2022. URL: <https://finbold.com/cardano-announces-the-doubling-of-rewards-for-hackers-to-spot-possible-vulnerabilities/>.
- [2] *Cardano is a decentralized public blockchain and cryptocurrency project and is fully open source*. URL: <https://cardano.org/>.
- [3] Manuel MT Chakravarty et al. “Hydra: Fast isomorphic state channels”. In: *Cryptology ePrint Archive* (2020).
- [4] Kyle Croman et al. “On scaling decentralized blockchains”. In: *International conference on financial cryptography and data security*. Springer. 2016, pp. 106–125.
- [5] John Day. *Patterns in network architecture*. Pearson Education India, 2007.
- [6] *Future hydra upgrade makes Cardano a speculative buy*. URL: <https://finance.yahoo.com/news/future-hydra-upgrade-makes-cardano-214314643.html>.
- [7] Evangelos Georgiadis. “How many transactions per second can bitcoin really handle? Theoretically.” In: *Cryptology ePrint Archive* (2019).