Blockchain and Cryptocurrencies Assignment - Report

Jack Leyland (dwmr15)

March 13, 2019

1.1 - Proof of Work

My User ID – dwmr15

The block target value in Hex

0x3e7fc180000000000000000000000000000000000000000000000000000

The nonce value of the valid block

553137

Number of double-hashes to find the valid nonce and the time taken

My method was to increment the nonce value by 1, double-hash the block and then check if the block is valid. Since my valid nonce value was 553137 and I started with the nonce value 0, this means I made 553138 double-hashes (or 1106276 single SHA256 hashes).

This took 9.2717 seconds.

Estimated time at Difficulty 1

My hash power is approximately $\frac{\#double\_hashes}{timetaken} = \frac{553138}{9.2717} = 59658.7465082$ hashes/s.

At difficulty 1, the target value is the initial target value:
0x00000000ffff0000000000000000000000000000000000000000000000000000.

This has 8 leading zeros in Hex, so 32 leading zeros in binary. This means that it will require approximately 2^{32} hashes to mine a valid block.

Using the earlier-calculated hash power, I can calculate how many seconds it would take to make this many hashes (and thus mine a valid block).

Seconds required $= \frac{2^{32}}{MyHashPower} = 71992.248369$ seconds

Estimated time at Difficulty 7,454,968,648,263

To calculate the target value at this difficulty, we use the equation $CurrentTarget = \frac{InitialTarget}{Difficulty}$. This results in: 0x25c1910000001580000000000000000000000000000000.

Target values are 64 hex digits which means that this target value (46 hex digits) has $18*4+2 = 74$ leading zeros.

Using the same idea as in the previous question, the seconds required to mine a valid block would therefore be $\frac{2^{74}}{MyHashPower} = 2.6238194 \times 10^{17}$ seconds.

1.2 - Proof of Stake

NOTE:, when checking these values in *verify.py*, please remove the space in the Hex string that I added so that the string would break onto a new line.

ECDSA public key in Hex

0xf13d1d92e0bffcdbee0309ada8f882572842e8b3f639704605872fde6770e552e26e62c5724c14a3deaa26 0593db03227d8830179529f73da8704a3e1abedc2c

Signature of 'Hello world' in Hex

0xe5a25870b1b53a5294761fcb52bf64ee163794ff43d288fcd5f593fa83b552a7862a0accf7cc5247f8a38b2 d13c68f82aedb6cb5067ee2944c7552a621276349

Signature used in calculating hit value in Hex

0x08d7cbf6883ca98981602dbd41380c02a3360c82e45922b9bda0be30da51caaa2c6162aae7778a30e1ac 768ac6b86a3deefb6ebabcda7fb5269fdfdb75c24b26

My hit value in Hex

0xd5aac7b8d1f04130

Time to forge a new block

For proof-of-stake, each account calculates its own target value, T, where $T = T_b \times S \times B_e$ (T_b=the base target value, S =the time since the last block in seconds and B_e=the effective balance of your account.)

We can mine a block when our Hit Value is less than T. Our Hit Value is calculated by signing the previous block generation signature with our private key, hashing the result, and taking the first 8 bytes of this result. This means, using the above equation for our target value, we can mine a block when $Hit\_Val < T_b \times S \times B_e$. We can rearrange this to find the seconds since the last block that we can mine a new block.

My effective balance is 15 (from my username). We can get the base target directly from a field in the previous block header. To calculate my hit value, I generated an ECDSA key pair, signed the previous block generation signature (found in the block header) with my private key, double-hashed the result and then took the first 8 bytes of this hash result.

By rearranging the equation, I can expect to mine a block after S seconds, where:

$S > \frac{Hit\_Val}{Base\_Target \times EffectiveBalance}$

$S > \frac{0\mathrm{x}d5aac7b8d1f04130}{1229782938247303*15} = 0.7843$

Which means that we can expect to mine a new block after **0.7843 seconds.**

2.1 - Interacting with the Bitcoin Testnet

My User ID – dwmr15

Transaction 1 - Newly generated coin

URL: https://chain.so/tx/BTC/57d5dd5915bf174eff72a4f3189e3b7d32a9e769817243dc72f86abe7b1b2a0c

This transaction is the first in block 566,015. This transaction represents the creation of new bitcoins; the coinbase transaction. All of the output goes to a specific address: (3KF9nXowQ4asSGxRRzeiTpDjMuwM2nypAN) and 0.0 bitcoins goes to a nonstandard address.

On analysis of the actual input and output scripts, we can see that the input was the 'coinbase', as we would expect as this is where newly generated coins come from. The transaction has a field 'received_from' which is null, since this is the first transaction of this block, the coins are generated rather than redeemed from a previous transaction.

The first output is to transfer these coins (the block reward) from the coinbase to the address mentioned above, who must be the successful miner of these new coins. The other output is of exactly 0.0 bitcoins to a nulldata address using the OP_RETURN operation. This marks the transaction as invalid, which allows the coin generation transaction to have extra data attached to it, in this case, a nonsense string: "?!??$?=???˜P??]? ???}? ?????????&!p"

Transaction 2 - Witness

URL: https://chain.so/tx/BTC/6d1687d24d2077f0b37cdc812cb676cf4cc9e20d44435aa9e91af15abb0b457a

This transaction has only 1 input but 11 outputs. Approximately 1.664 bitcoins are being sent from one input address to 11 different outputs. Furthermore, this transaction has witness data in the input script; neither of the other 2 transactions that I have described have any witness data.

The transaction type is a Pay-to-script-hash, as we can tell from the structure of the majority of the output scripts: 'OP_HASH160 <20-byte-hash> OP_EQUAL'. The output script for output number 8, however, is a pay-to-address script.

Transaction 3 - Same input and output address

URL: https://chain.so/tx/BTC/09f752fd0ffff026d6d977ea1fa8243072a83b4edc299336b377f574f38e7156

This example is a strange transaction. There are two input addresses to the transaction, totalling up to approximately 1.6873 bitcoins, with the vast majority (all except 0.00000456 bitcoins) coming from address 12fVdGzZpVoe4E9MUtCzXa4NNjZ6rTGKLL. This most likely means that one user is spending some bitcoins, but to do so he must put together bitcoins from different past transactions. Interestingly, the output is all sent to this same address (as two separate outputs), along with a 3rd output script using OP_RETURN to attach some additional data (a message) to the transaction.

My theory is that one user is trying to gather all of his bitcoins into one of his addresses, but once fees are deducted he has actually made a loss, with the total output being approximately 1.6870 bitcoins.

This transaction is a pay to address transaction, as we can tell from the output scripts being in the form 'OP_DUP OP_HASH160 <Pub key hash > OP_EQUALVERIFY OP_CHECKSIG'.

2.2 - Sending 100 Satoshis

My Bitcoin testnet address

mo94jePN64JaJPXa87SQQqxUT1SbgKcE66

Transaction ID

71233edb0a780d3ade9415b3273952d7b45d50dd685474d52b86d7ece79a5226

Transaction URL

https://chain.so/tx/BTCTEST/71233edb0a780d3ade9415b3273952d7b45d50dd685474d52b86d7ece79a5226

2.3 - Proof of Burn

Transaction ID

d89e26fef66caafa896b9e7c3fed2b29460b6bda21149f4d04e9861662bfb6df

Transaction URL

https://chain.so/tx/BTCTEST/d89e26fef66caafa896b9e7c3fed2b29460b6bda21149f4d04e9861662bfb6df

Script in Hex

0x6a4c06746573746572

How I put the script together

To put my script together, I first took a look at the example proof-of-burn transaction provided. We specify the script in the outputs of our transaction, so I looked through the raw transaction data and located a hex script in the output where the value being sent was 0. This is the script we need to replicate.

Firstly, as in any proof-of-burn transaction, we need to start with OP_RETURN (0x6a). Then, as in this example transaction (and as mentioned in the Bitcoin Script Wiki), we need to follow this up with an OP_PUSH_DATA1 operation (0x4c), to indicate that we would like to attach some data to the transaction. We indicate in the next byte how many bytes we are attaching (0x06, since our username hex encoding is 6 bytes long), and then the hex encoding of our username itself (0x6a4c06746573746572).

Putting all of this together creates a valid script as you can see above and inspect at the URL provided.

3 - Investment in BTC, NXT or Gold?

I will recommend the investment of Gold, on the basis that it more appropriately holds key properties of any currency that you would want to invest in, and eventually spend/exchange in the future.

One of the most important of these properties is that your currency can be used to **store value** until it is needed. Gold holds a relatively consistent value, only varying by approximately 28% in the last 3 years between a maximum and minimum values of £1060 and £827 respectively. 1 BTC was worth approximately $20K at the start of 2018 but is now only worth $4K, an 80% decrease in value, and NXT has followed a similar trend, falling from $2.14 to $0.15 per coin in just the last 3 months. The volatility in the value of these cryptocurrencies provides enough of a risk to disuade investment in them.

Secondly, we want our currency to be **accessible and profitable**. Bitcoin mining requires significant computational power. With a standard CPU setup achieving approximately 1MH/s, it would take millions of year to mine even one Bitcoin block. Alternatively, if we bought an ASIC, making up a larger proportion of the hash power we could mine more successfully. However, the energy required to run such hardware is higher than the average block rewards you would achieve, completely nullifying the possibility of investing in Bitcoin unless you can become part of a mining pool - however this contributes to the centralization of Bitcoin which goes against the very purpose of the cryptocurrency. Contrary to this, we can instantly purchase Gold online extremely easily (e.g. bullionbypost.co.uk), making it much more accessible than Bitcoin.

NXT's major downfall is that its market cap is currently only $25million. Therefore, even if we were to own every NXT coin (in which case the currency would die out), we are limited to our returns. Compare this to Gold's rumoured $7 trillion market cap; there Gold has the potential to be much more profitable than NXT.

Finally, you would ideally want investments that can be **readily exchanged** for goods and services. Gold dealers exist everywhere, especially online, where we can readily exchange Gold for GBP (or any Fiat currency) which can be spent anywhere. Bitcoin has a vastly technical underpinning which complicates transactions and exchanges, reducing its exchangability. Although there is increasing support for spending Bitcoin (e.g. Overstock.com), you cannot say Bitcoin can be readily exchanged. NXT is synonymous here, increasing but limited support which does not come close to the readiness of Gold.

Furthermore, Gold can be **held physically**, whereas with Cryptocurrencies we have a record tied to our online address. This is a much more unreliable store, as simply forgetting your password to your online wallet could lead to you losing your entire investment, as one user found out, losing $30,000 with the wallet service Trezor.

These key properties combined lead me to conclude that Gold would be the most appropriate for investment.