Manual Blockchain Puzzle Tournament

Sam Clark, Ryan Estes
Plaid Lab, University of Vermont

2020
December

Abstract

In typical blockchain applications, miners race to solve a hard puzzle in order to receive a reward for adding a block to the blockchain. This puzzle involves finding the solution to an algorithm that proves that the miner did some work involving the block to be added. We propose a method for shifting the responsibility of solving this hard puzzle to human users, and removing all direct computer networking in favor of verbal communication. We present our method in the form of a puzzle competition for human players.

1 Introduction

In typical blockchains applications, a miner will first decide on a set of contents they wish to turn into a block and add to the blockchain. Next, in order to show a proof of work, the miner searches for a value (known as a nonce) which, when combined with the main contents of the new block, hashes to a value satisfying some condition such as being less than a given value. The minor also adds a piece of information which identifies their account so that everyone knows who gets the reward for mining this block.[1] This means that each miner is solving a slightly different puzzle, unique to their account identifier.

Our protocol mixes things up by changing the proof of work from solving a hash to solving a word puzzle. Miner's are now known as contestants and are working to earn points for themselves. Each word puzzle is generated from a seed which is on the previous block or a random, agree upon value in the case of the first round, and the miner's id. Each contestant's computer will take their id and the current seed and shuffle it in a reversible way. This shuffled value is then used to generate a word puzzle. Only the puzzle itself is revealed to the contestant. The solution to the puzzle is the shuffled value used to generate the puzzle, meaning that this protocol is not secure in the malicious case.

Upon solving the puzzle, the contestant will know the shuffled value that when unshuffled will equal the original seed and that contestant's id. The contestant shares this value with the other contestants who can each ask their computers to verify that it is a valid solution. Upon doing so, the unshuffled value containing the winner's id is added to the blockchain. A new seed value is obtained based deterministically on the block that was just added. Importantly, the new seed depends on the id of the winner. Everyone drops what they are doing and begins a new round with the new seed.

Once the contestants agree that the game is over, a winner is decided by counting the number of times each contestant's id shows up in the blockchain. The blockchain is validated by ensuring that each block's seed portion is based on the previous block, and that the first block's seed portion matches the agreed upon starting seed. For additional security, contestants should also ensure that their local blockchains match each other's. In order to fake a single block in the middle of a blockchain, a semi-malicious contestant (one who breaks the rules but does not collude with their own computer or know the shuffle function) must, in addition to solving their version of the puzzle for that block, solve someone's puzzle for each subsequent block up to the most recent block. Since each puzzle is based off of the winner of the previous round, a semi-malicious contestant cannot reuse the work done by other winners.

2 Protocol

- All contestants agree on the number of maximum players as a power of 2 (recommended no more than 2^4, absolute maximum is 2^{15}). So the max number of players $= 2^n$ for some n between 1 and 15 inclusive.
- All contestants agree on an arbitrary starting seed of length $(16 - n)$.
- All contestants agree on each contestants unique unique ID such that $0 \leq \text{ID} < n$.
- Each contestant's computer takes their ID in bits and combines it with the starting seed to form a personalized 16-bit starting seed.
- Each contestant's computer will shuffle these bits around using a public (but inaccessible to contestants), reversible algorithm, and uses these new bits to generate a word puzzle.
- To generate a puzzle, each computer splits the shuffled 16 bits into 4 nibbles. It uses the first three nibbles to look up 3 words from 3 public, 16 entry long ordered tables of English words. Then, it encrypts the 3 words using the last nibble as a caesar cipher key (note that there are only 16 possible values for a key rather than the normal 25, but that is ok).
- The 3 encrypted words are then presented to the player, along with 3 reference cards containing all the possible words, indexed in order using hexadecimal numbers from 0 to F.
- The contestants each work to figure out the 3 hexadecimal values corresponding to the original 3 words as well as the key, which also needs to be translated into hexadecimal.
- Once a contestant has come up with a solution, they notify the other players that they think they have solved the puzzle. They shout their solution, like this: "Hey everyone! F73D!".
- At this time, each contestant (including the one who shouted their solution) stops what they are doing and enters the solution into their own computer.

- Each contestants's computer converts the hexadecimal value back into binary (if valid, this value should exactly match the value that was used to generate the puzzle of the player shouting the solution).
- Each contestant's computer runs the shuffle algorithm on the value in reverse and validates that it is a correct solution by checking that the (16 - n) seed bits match exactly.
- If the computer reports that the code is invalid (the contestant did NOT shout a valid solution) all players carry on trying to solve the current puzzle.
- However, if the solution is valid, each contestants's computer reads the n-bit ID portion of the unshuffled solution and reports that the player with that ID just won a point and the current round is over.
- The valid solution is pushed to the blockchain, and a new (16 - n)-bit seed is generated using the solution just pushed in some simple public (but inaccessible to the contestants) algorithm.
- A new round starts with the new seed, and the process is repeated an agreed upon number of times.
- Once everyone agrees that the game is over, points can be tallied by running through the blockchain and counting how many times each player's id shows up.
- Verification can also happen at this step, since any discrepancy in winning IDs will cause a fork, which can be detected by a majority of honest players comparing results.

3 Conclusion

Our protocol allows users to compete in a puzzle tournament using only their voices to communicate. It is secure against semi-honest adversaries, as well as malicious adversaries versus an honest majority. Just like in normal blockchain applications, a verification step can be performed to detect faked blocks in the chain. Our protocol can be adapted to all sorts of different kinds of puzzles though it would take some care to do so. Future work might be to come up with a way to generate a puzzle in such a way that the solution is different from the values used for generation which still requires a human to solve. Maybe some kind of image recognition puzzle.

References

[1] https://en.bitcoin.it/wiki/Block