Quantum Blockchain

Adriana Palos, Pablo Veganzones, Pablo Viñas, Gabriel Escrig

November 18, 2021

Contents

1 Introduction

Traditional Blockchain systems are a great way to protect and store information without depending on a central authority. However, to maintain decentralization, it is necessary to figure out a method to "randomize" which node has the privilege to impose the order in which information is stored after a period.

This method is usually known as proof of... (whatever). It plays a fundamental role because is thought in a way in which the privileged node is forced to demonstrate its unique identity, so that he/she doesn't have a greater chance to be chosen than what he/she deserves. If the work is done properly the node will receive a reward.

The problem is that these proofs of uniqueness are not perfect. Proof of work, the most widespread method of its kind, is also a fantastic way to waste computational power, and thus, resources. Proof of stake, another commonly used method, gives more chances to those who participate with more funds, so that rich nodes get richer. Is there a way in which Quantum Computing can solve those problems? It turns out it does.

In this project it is shown a way to make a Quantum Blockchain from states in the block sphere. In section 1 it is shown how the classical blockchain works. Section 2 states the principles to understand how the quantum blockchain works, which is explained in section 3. Section 4 shows the problems that arise when implementing it in Qiskit and a discussion of the solutions. Next, conclusions, but also some ideas we came out with to apply in a further study or hackathon are shown in section 5, as well as how it would need to be implemented.

1.1 Classical blockchain

A blockchain is a growing list of records, called blocks, that are linked together using cryptography. Each block contains a code of the previous block and all the transaction data. This code is the well-known hash function, and it is a mathematical algorithm that maps data of an arbitrary size to a bit array of a fixed size. In a simpler way, hash function can be visualized as any kind of function which maps arbitrary information to a set of values. When those functions are complicated enough, like SHA-256, one can treat them as irreversible. As each block contains the hash generated by all the data of the previous block the connection between one block and the following is univocal, so they form a chain in which each additional block reinforces the previous ones.

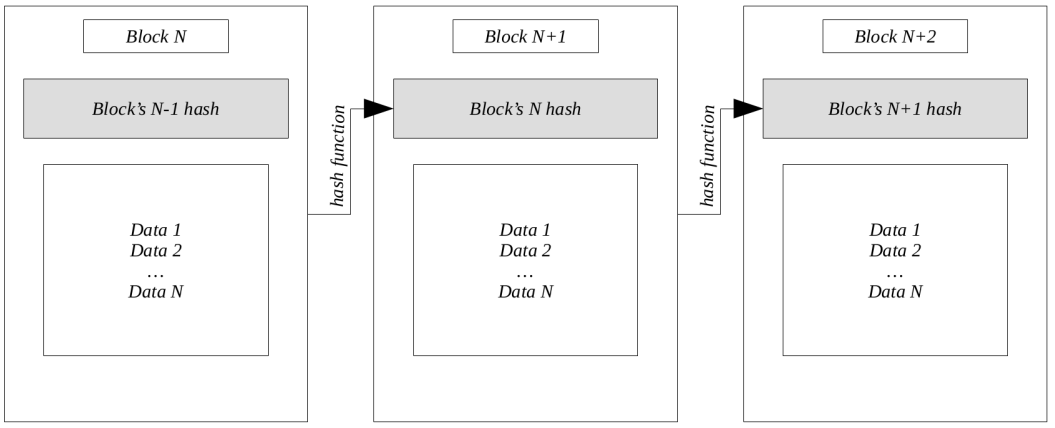

Figure 1: *Classical Blockchain*

2 Main concepts

2.1 Hash-states

One can use Hash functions to hash almost anything if the process is thought carefully, so one question arises: why not hash a quantum state? Indeed, an arbitrary state in the block sphere can be described as

$$|\psi\rangle = cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}sin\left(\frac{\theta}{2}\right)|1\rangle \tag{1}$$

Taking that into account, one just have to take the *block* of information, divide it into two pieces, and use the hash function to generate the parameters θ and ϕ. Then one can generate a Hash-state.

2.2 Fidelity

Although most readers will be familiar with this concept, it is important to remark its importance in the process, as it will be widely discussed later. Fidelity between two states gives an insight into how close those states are. It is useful to think about it in terms of distances in the Block sphere.

Given two states $|\psi_a>$ and $|\psi_b>$, fidelity is defined as:

$$F(\rho_a, \rho_b) = \left(tr\sqrt{\sqrt{\rho_a}\rho_b\sqrt{\rho_a}}\right)^2 \tag{2}$$

Where $|\rho_i>=|\psi_i><\psi_i|$ are density matrices.

3 Quantum Blockchain

3.1 Teleportation and fidelity between states

One important element for our Quantum Blockchain is teleportation of states. Once each node has computed its state,

$$|\psi_i > \textit{ with } i \in N, \textit{ where } N \textit{ is the number of nodes}$$

then needs to share it with every other node. Due to the no-cloning theorem every pair of nodes needs to teleport their state to make the blockchain possible. That means that node 2, for example, needs to share $|\psi_2 >$ and will have $|\psi_i >$ different states.

Analogously, every node has N-1 states plus the one they generated. The way this quantum blockchain works is by comparing fidelities between states, which references to how much alike are the states. Deeper explanation can be found in sections 3.3 and 4.2.

3.2 Proof of fidelity

Lets assume each node receives information in its own order. They could then agree on the fact that a block is closed when received a fixed amount of information, i.e. three transactions. At that point, each node would generate a Hash-state with the information contained in its block. How can we establish a privileged node?

Well, suppose that nodes are connected both by a classical and a quantum channel. In that case, they all can send their Hash-states to the rest of the chain, through teleportation. At this point, one has to figure out the "randomizer" method. A simple way to achieve that is by measuring the fidelity between the states received by a node and its personal Hash-state. If all the nodes repeat that same process and send the results through the classical channels, a node would have sent a state which, when compared with another node's state, gives the maximal fidelity between all the states compared. The node who has sent that precise state would be our privileged node.

Up to this point we have developed a clever method to select nodes randomly. The question is, why would it have any kind of benefit compared to classical ones? We have said that nodes are connected through quantum channels. If the process itself of establishing a quantum channel is costly (and today definitely it is, for example the act using teleportation itself), nodes won't be able to simulate that they are more than just one, as they would have to build new quantum channels. Thus, imposing the necessity of sending quantum states using teleportation to be a verified node makes the process fair. Then one has all the great advantages of wasting computational power without actually wasting computational power.

3.3 Quantum Blockchain scheme

With all the explained above, the quantum blockchain works as follows: with the maximum fidelity of the earlier block, the transactions and a random number the hash function generates two parameters that determine the state of each node.

With that state and the ones generated by the other nodes, and shared through teleportation, each one calculates all the possible fidelities (25 for each node in a system with 5 nodes). These fidelities are checked by all nodes.

If the values are exactly the same the system validates the block and the node whose state has the maximum fidelity mines, and therefore gets a reward. Because they have build a valid block in the first attempt, then all the nodes get a maintenance reward (see section 4.3). If the values differ then the system must start all over and will not get the maintenance reward. Once the system validates a block, the maximum fidelity is passed onto the next block and the next block formation begins.

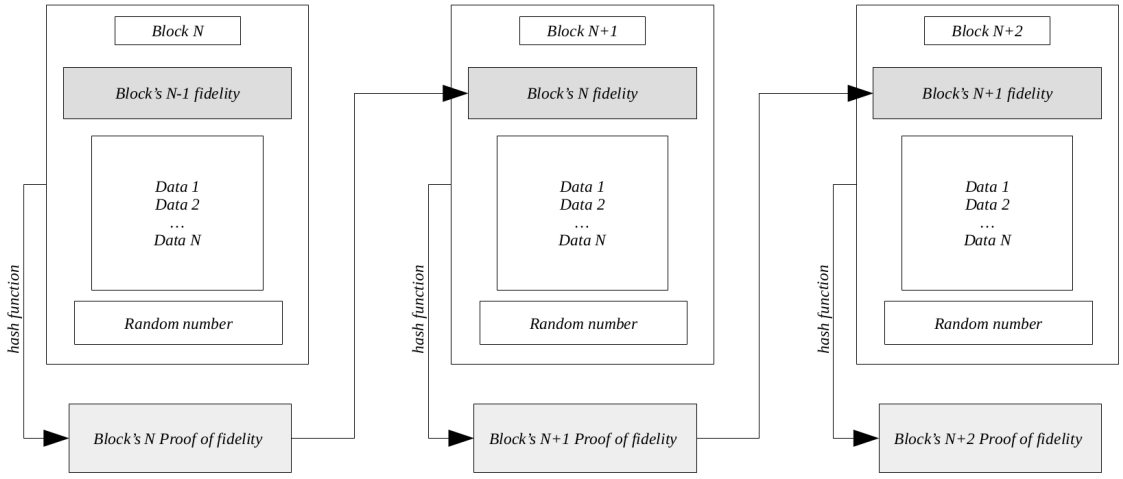

Figure 2: *Quantum Blockchain*

4 Potential problems and solutions

In order to make proof of fidelity an useful method it is necessary to solve some subtle issues. Those issues are discussed in this section, and a solution is given for each of them.

4.1 Determinism

The first problem arises when a node realizes Hash-states are completely deterministic. One could try to predict the order in which transactions are going to be sent to a node.

If that happens, the cheater node would know the state generated by its partner's block, and therefore, he/she could send a state as close in fidelity as wanted in order to be the winner node.

This problem has a simple but efficient solution. Apart from the quantum state, each node has to generate as many quantum numbers as nodes in the chain. All nodes will then send all the random numbers so that these numbers must be included by each node in the final blocks. One can not guess all random numbers so that blocks become unpredictable. And, isn't quantum computing a good way to generate random numbers? We have followed the work explained in [1] to generate random numbers using quantum circuits of the form

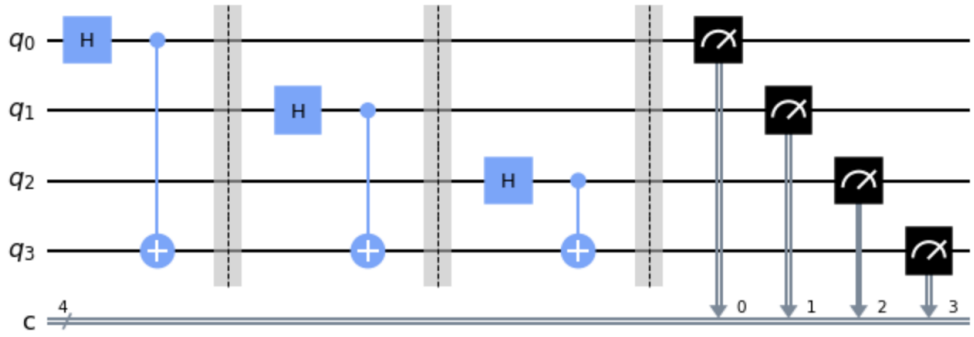

Figure 3: *Quantum circuit for random numbers*

4.2 Symmetry of the process

You may well have thought that, whenever the winner node is determined, there are two nodes with the maximum fidelity, as when two nodes measure fidelity between their respective states they get the same result (fidelity is symmetric). This is true but, in reality, it does not represent a real problem. Rewards can always be given to both winners, and, to determine the correct block, it is only necessary for the chain to agree on the same criteria of election.That could be to choose whichever node lives further east, or whichever node has won less times, or any other random consensus. In our code we are giving the privilege to the node defined first, but it is just a matter of simplicity.

An alternative solution could be related to changing the state used to measure fidelity so that it is different from the one sent. In that way, the process is not symmetric anymore.

4.3 Verification

The process is now unpredictable, but, what if a node decides to be malicious and sends a state which differs from the one that its block generates? That's easy to solve.

As all nodes know how hash-states are built, they just have to compare the fidelity of the theoretical states generated by both the blocks sent and their own block with the one they got from the received states. Thus, it is only necessary to include sending the blocks to the rest of the chain as a part of the process to detect a malicious node.

For example, suppose our system has 5 nodes. Then by calculating every possible combination, each node would have 25 fidelities. For node i, the fidelities he calculates would be F_{ji} with $0 < j < N$. Note that $F_{ii} = 0$ otherwise it would mean node i compares $|\psi_i >$ to $|\psi_i >$ obtaining always a fidelity of 1. The fidelities he must confirm are F_{jk} with $0 < j < N$, and $k \neq i$. This way, each node must come up with a table like this:

node 1	node 2	node 3	node 4	node 5
F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
F_{21}	F_{22}	F_{23}	F_{24}	F_{25}
F_{31}	F_{32}	F_{33}	F_{34}	F_{35}
F_{41}	F_{42}	F_{43}	F_{44}	F_{45}
F_{51}	F_{52}	F_{53}	F_{54}	F_{55}

Table 1: Fidelities involving five nodes

In that case the only additional condition is to agree on the fact that only completes a block if all fidelities are verified, which means that all five tables like the shown above are identical.

Obviously, for the verification process to be complete, it is also necessary to check that the random numbers in a block are the right ones (which is easy because each node knows his random numbers).

4.4 Maintenance reward

If all nodes agree on the fact that only verified blocks (in terms of fidelity and random numbers) are valid, what prevents a node from sending the wrong state just because he/she does not want the block to be validated? It is necessary then to introduce a "maintenance reward". As in a traditional Blockchain, the winner node receives a reward, but, in addition, if the chain itself works in the right way, all nodes will receive an additional small incentive, just to make sure it is not worth it to act maliciously.

5 Conclusion and future lines of investigation

Through this work, we have reviewed the main concepts of our Quantum Blockchain system and the new Proof of Fidelity. Probably the most important thing to note is that the system is consistent in terms of security without wasting a vast amount of computational power, making its carbon footprint much smaller. Also, that improvement is a property derived from the quantum nature of the process.

As a future line of investigation, as a team, we have discussed the scalability of the project to a larger number of nodes. Some minor changes should be made in terms of the information connecting blocks (maybe not using all fidelities), just to make the system more efficient. The system is perfectly valid if not all nodes are connected to all the others, but the implemented code would need to be generalized.

Also, we note that it is extremely difficult to find a block which generates the same exact winner fidelity to the one of a previous one. Anyway, to make that even more difficult (it would be almost impossible to achieve), the system could be generalized using two states per block, thus, one could increment the number of fidelties involved and that would lower the probability of finding an equivalent block, as it would have to generate two equal fidelities instead of one (or even more if wanted).

We are conscious about the fact that using teleportation between blocks may seem very optimistic nowadays. Nevertheless, it establishes a new way to verify node's identities, just because it is not a trivial process to achieve.

Finally, having described a system in which quantum states are sent with ease, it seems natural to think about the potential of storing quantum information in the blocks and how would that affect to the unmodifiable nature of the chain.

6 Bibliography

- Satoshi Nakamoto.*Bitcoin: A Peer-to-Peer Electronic Cash System,* `https://bitcoin.org/bitcoin.pdf`
- `https://qiskit.org/textbook/preface.html`

References

[1] Janusz E., Jacak, Witold A. Jacak, Wojciech A. Donderowicz & Lucjan Jacak. *Quantum random number generators with entanglement for public randomness testing.* (2020).