Intangible

A low energy blockchain

Ricardo Schiller, Nuno Rodrigues
Agap2IT - May 2016

Abstract
Blockchain technology has brought a method for distributed consensus on an history of operations without the need for a central administration responsible for assuring and maintaining this consensus. One of the pillars of blockchain technology is the concept of proof of computation work, required to ensure the security of the system, which is incentivized by the possibility of a remuneration of digital coins, an activity called coin mining. The most prominent blockchain is the Bitcoin blockchain. On Bitcoin, the proof of work is periodically adjusted to ensure that the mining of blocks remains at a constant pace. The incentive of mining Bitcoins, by overcoming the proof of work challenge, started an unprecedented race for Bitcoin mining, where specialized hardware was created and mining pools were organized, which lead to performing today approximately 1.5 exahash per second by the whole Bitcoin network, an effort that consumes an enormous amount of energy. The main goal of the proof of work is to provide security to the system, but on the other hand is also leading to centralization through due to large mining operations. Here we propose Intangible, a new approach to the Bitcoin blockchain algorithm, which has a different consensus algorithm, that dramatically reduces the computational effort required to create a network with equivalent security to the Bitcoin network, and that mitigates selfish mining, mining centralization, computational waste on sidechains, and 51% attacks.

Introduction
Traditionally, the required proof of work to produce a candidate block is chosen in accordance with the estimated average time intended for overcoming that proof of work. On the Bitcoin network, the proof of work is regularly adjusted to ensure that a block is mined, on average, every 10 minutes. The race to Bitcoin mining required an increasingly difficult proof of work, today being necessary to possess specialized hardware to have some chances of successful mining. This translates to infrastructure centralization, something that wasn't intended on the original concept of Bitcoin. Alternative proposals exist to the proof of work method, designed to be bound by CPU power, IO capacity, memory capacity, and other factors. Other proof types have been suggested, e.g. proof of burn and proof of stake. Nonetheless, none of these methods completely solves the problem, opening up other possibilities for centralization.

To solve this problem, we propose an algorithm that has some key differences from the traditional blockchain algorithm. Here, we assume that a chain grows at steady intervals of time T, independent of proof of work. Proof of work and block time are chosen such that a miner with few resources can overcome the proof of work inside the block time. It will be a considerable effort for a low budget miner and minimal for a large budget miner. Nonetheless, the next block will be added only after a period of time T, since the last block. After Thas passed, the phase for the selection of the next block of the blockchain, begins.

Choosing the best pair - The Pairing Algorithm

Upon reaching block time, most miners will have a candidate block ready to present. Instead of just choosing the block with highest proof of work, a different approach is taken, that has two phases, where the first goes as follows:

- Every miner needs to broadcast his candidate block to the other miners
- Having received all or most candidate blocks, every miner produces all combinations of concatenation of pairs of proofs of work from every candidate block
- For every concatenation of pairs of proofs of work, a hash is calculated from the concatenation

In this sense, the maximum number of hashes that each node must perform in order to calculate its candidate for the next block is expressed by:

$$A_2^n = \frac{n!}{(n-2)!} = n.(n-1)$$ where n is the number of candidate blocks

For example, considering three miners, each with its own candidate block and its corresponding proof of work, each one broadcasts his candidate block and upon receiving candidate blocks from other miners, starts trying out all combinations of proofs of work, obtaining the following set of possible pairings:

$$\{\, PoW1 \,||\, PoW2\,, PoW1 \,||\, PoW3, PoW2 \,||\, PoW1, PoW2 \,||\, PoW3, PoW3 \,||\, PoW1\,, PoW3 \,||\, PoW2 \,\}$$

From this set, the miner calculates the hash of each pair, obtaining a list of hashes. Considering the set of possible hashes (2^{256} hashes for sha256), we can imagine the calculated hashes being distributed evenly along the space of possible hashes, from lowest to highest. We are looking for high hashes, they will help us select the pair containing the winning block. The main reason for it, is that from a network of $n$nodes we get n^2 (almost) of hashes. If an attacker wanted to get a hash as high as the highest hash of the network, he would probably have to try out a similar amount of pairs of proofs of work to obtain one. An attacker with strong computational power could generate by himself many different candidate blocks and test out all combinations of proofs of work to obtain hashes even higher than the network. This is dealt with on the second phase of the pairing algorithm.

Miners are people, not farms of pools

If we considered high hashes only, as the way to determine the next winning block we would be in trouble. The fact that the proof of work isn't hard for serious miners, and that new blocks are added to the blockchain at a predefined intervals, an attacker could build a sidechain easily, provided he had computational power to produce around n candidate blocks with corresponding proofs of work and calculating all the pairing combinations. To render useless this attack, the following rules are added:

- The pair of candidate blocks that produce a hash must have different mining addresses from each other
- The mining address of each block should be linked to a person, and a person should have only one mining address
- The identity of a person should be confirmed through external authorities
- Each confirmation from an external authority should have a weight that is agreeable by every node (hardcoded on the source code of the application, for example)
- A winning block can't contain a mining address that belongs to an identity that has *recently* mined

To calculate the winning pair we use the confirmation weight c and the hash size h to find out the pair that has the highest value for the equation:

$$Max(\alpha c_i + \beta h_i), \text{ for every pair } i$$

where $\alpha$and $\beta$are predefined constants that balance the confirmation weight and the hash size. We consider a high hash more important than a high confirmation weight, so we could state that α and β should be such that $0 \leq Max(\alpha c) \leq 1 \; and \; 0 \leq Max(\beta h) \leq 2$. This just means that the hash size will have twice the importance of the confirmation weight, nonetheless, $\alpha$and βcould be better adjusted by further analysis.

Now that we have a winning pair, a last check is made. The winning pair can't contain a block with a mining address that is present on a recently mined block. Taking into account that the pair selection is a random process, and provided there is a strong identity, having many consecutive blocks mined by the same individuals, on a network with many miners, clearly states that something is wrong. If the winning pair has mining addresses belonging to identities that have recently mined a block, the winning pair is discarded, and the next best option is chosen.

The winning block is the block that has its proof of work as prefix on the concatenation of the pair that has the maximum value according to $Max(\alpha c_i + \beta h_i)$, and that has no mining addresses on recently mined blocks. The winning block should have appended the information of the proof of work of the pair block, the corresponding mining address, the pair hash and the identity scores. The mined coins are distributed evenly between both addresses of both miners.

The way to determine what is considered recent mining is straightforward. We need to calculate the probability of successfully mining a block on a network of Mminers after n tries. Given enough tries, this probability tends to 1. A probability threshold has to be set that determines how many blocks must pass before a new block mined by the same individual is allowed.

As we can see, requiring a strongly confirmed identity for a miner is crucial to the success of this algorithm. Identity confirmation is an old problem, also present on the web, being the DNS a reference in this area. Identity confirmation as presented here has already been implemented on blockchains, by various companies. Onename is a very good example of

this, built by Blockstack Labs. This identity confirmation system was developed to be agnostic of the blockchain platform where it resides, so it's not unreasonable to consider a for to this blockchain. Nonetheless, identity is only required for miners, the remaining users may stay anonymous.

Do we still need proof of work

If we imagine a blockchain where every miner is well identified, and where sequential blockchain mining by the same miner isn't allowed, one could ask if proof of work still remains important. We think it still is important for at least one main reason. Blockchain mining should require some effort. Miners should be aware they are investing in something important, that will give them an eventual reward in coins. If no proof of work was required, the cost of participating in the network was virtually none, which could produce massive changes to the network topology and lead to scalability and consensus problems.

The scalability problem

One of the major problems of blockchain technology is the ability to storage it over time. The size limit of each block combined with the mining rate, determines the maximum size the blockchain will have after a certain period of time. One way to reduce the size of a blockchain is by pruning it, producing a reduced version of the blockchain with only unspent transactions, leaving the responsibility of archiving the entire blockchain to a small set of network nodes. The problem with this solution is that it tends to reduce the number of nodes having the whole chain, which in turn leads to centralization. Intangible has the same problem, plus the following one: as each node has to find the best combination of blocks between all the blocks broadcasted over the network, first it needs to receive them - which requires bandwidth -, and second it needs to store them temporarily to perform all possible pairing combinations. As more miners participate on the network, bandwidth and memory become a bottleneck.

Sidechains to the rescue - Forks and Merges

Assuming the number of miners from the Intangible network can grow to a much larger number than the bitcoin network, we need to consider what would happen when more and more miners adhered to the network. Eventually, miners would start to experience lags upon receiving candidate blocks, probably before worrying about memory issues. Consensus would fail more often, and sidechains would appear. We propose to balance this issue by adjusting the proof of work to a higher value. Miners would be driven to assess if the eventual reward of Intcoins (Intagible currency) would surpass the cost of mining until then. By raising the proof of work, the number of participants should decrease. But what about all the other potential miners? We propose to solve this problem using special sidechains. The idea is as follows.

First of all we need to deal with the main chain, which needs to be maintained during the lifetime of the blockchain. For this purpose, we have to carefully choose a maximum block size and block time that together permit the chain to grow for decades. We mean blocks much smaller than 1MB and block times much higher than 10 minutes. This chain wouldn't have a throughput nowhere similar to Bitcoin, for example, but let's consider this possibility for a moment, along with the concept of special sidechains. These sidechains are intended to have different properties from the main chain, for example, large blocks and small block times. Contrary to the main chain, these chains don't mine new Intcoins. Instead, they will

recover invested coins. This is done by using a special block, called the sidechain genesis block (figure 1), that contains a transaction with a list of inputs from the the main chain, and an output that is the sidechain genesis block. This transaction is an investment on the sidechain, with a promise that these coins will be recovered by the miners at an established rate. The genesis block states all the rules of the sidechain: block time, block size, minimum proof of work, incentive per block, etc. We can imagine this genesis block as a fork from the main chain.

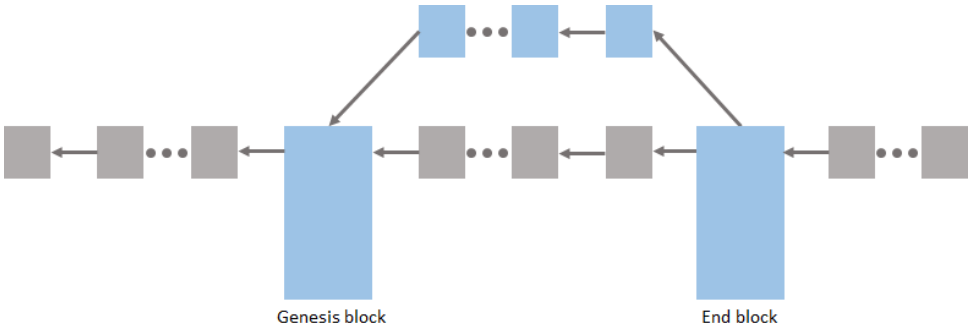

Figure 1 - An example of the start and finish of a sidechain through its corresponding genesis and end blocks

All addresses registered on the sidechain must be previously registered on the main chain, and should point to the block on the main chain where they were originally registered. The main reason for this is that, contrary to the main chain, the sidechain must have an end. When there are no more coins to retrieve, it is necessary to merge the sidechain with the main chain. For this purpose, an end block is written, transferring all unspent transactions out of the sidechain to the correspondent addresses, on the main chain. On the main chain remain only two blocks related to the sidechain, the genesis block and the end block. For sidechains where the history of transactions isn't important, the remaining blocks of the sidechain may not need to be maintained, and may eventually disappear entirely.

A world of sidechains

According to the rules of a sidechain, the existing miners may decide whether it is worthwhile to mine its blocks. If the reward per block is large, the number of miners will tend to increase, to the point of becoming a problem of scalability again. The investors of the sidechain can send a transaction that changes the rules, e.g. raising proof of work, to balance out the number of miners. These sidechains can be used to provide a specific service for a group of stakeholders. They can be used to bring more powerful players to the mining arena, with good rewards, enabling more competitive sidechains. Nonetheless, eventually, the the sidechain ends and merges to main chain again, when a new one may always appear, in the future.