SpaceMint: The Future

Ashwin Ignatius
Georgia Institute of Technology
aignatius3@gatech.edu

Marcus Loo
Georgia Institute of Technology
mloo3@gatech.edu

Gabor Siffel
Georgia Institute of Technology
gabor@gatech.edu

Abstract

Bitcoin uses Proof of Work in its block generation scheme that has proved to have drawbacks due to its high energy demands. This has given rise to unintended consequences such as being less friendly to small miners and the network becoming more centralized. SpaceMint is a cryptocurrency that aims to solve these problems by using a Proof of Space approach, an alternative scheme that relies on disk space rather than computational power. In this paper we analyze Proof of Space, SpaceMint, and their practicality.

1. Introduction

The history of the blockchain can be dated back to 1991 when Stuart Harber and W. Scott Stornetta formulated a secure chain of blocks with the purpose of having a place to store documents where the timestamps could not be tampered with, and a year later, they incorporated merkle trees into their design to increase efficiency [7]. Satoshi Nakamoto conceptualized blockchain in 2008, and Bitcoin was officially launched to the public in 2009 [7], and it has become one of the most popular cryptocurrencies with there currently being over 44 million blockchain wallets [10].

Mining in Bitcoin works through a Proof of Work scheme. Miners try to find a nonce such that when the block is hashed, the hashed value is below a certain difficulty threshold. The miner will then be able to add a block to the blockchain. Miners are rewarded through bitcoin rewards and transaction fees. According to Digiconomist, miners use a total of 71.64 TWh anually, which is comparable to the power consumption of the country of Austria [1].

Some of the current concerns over the Proof of Work scheme as illustrated through Bitcoin is that an individual has little incentive to expend its spare CPU cycles to mine for blocks because the cost of energy as a result of mining would be greater than the expected reward from mining. This stems from the fact that a lot of mining is currently done on specialized hardware called ASICs that is only used for mining, and a lot of the hash power is from a handful of mining pools [2]. If a mining pool obtains a hash rate of at least 51%, it could potentially undermine the security and stability of Bitcoin [9].

As a result, there has been interest in other types of schemes such as Proof of Stake and Proof of Space, which seems to help fix some of the problems that was addressed with Proof of Work, but at the same time seems to cause other issues. This paper will focus on a cryptocurrency that uses Proofs of Space called SpaceMint, its benefits, and how it aims to solve some of the issues that arise from using a Proof of Space scheme.

In this paper we analyze Proof of Space and its benefits and downsides compared to Proof of Work scheme. We explain the user-incentive issues that arise in Proof of Space based blockchain networks, especially of the "nothing-at-stake" problem. We summarize the implementation details, solutions, claims, and evaluations of the cryptocurrency SpaceMint, presented by Park et al. [8] Finally, we discuss possible issues that could affect SpaceMint's effectiveness. We draw our conclusions based on this analysis and existing storage-based cryptocurrencies to determine the viability of Proof of Space and SpaceMint in the real world.

2. Proof of Space

Proof of Space is a protocol where a user dedicates a certain amount of disk space to the service they are requesting. Both *Proof of Space* (PoS) and *Proof of Work* (PoW) are based on the idea that a requester of a service must invest a non-trivial amount of effort to show commitment to the service [5]. In PoW, effort is measured in terms of computational power, which can become very expensive. In PoS, effort is measured by disk space allocated. A key assumption of PoS is that users will have sizeable, unused amounts of disk space, which makes PoS essentially free [5]. By avoiding the expensive computations of PoW, PoS prevents the formation of a "mining oligarchy" and provides opportunities to weaker miners.

When implemented, PoS is a two-phase protocol between two parties, a prover and a verifier [8]. The first phase is *initialization* and the second phase is *execution*.

1. *Initialization* In the initialization phase, the prover will store some data S_γ of size N and the verifier will then store a short commitment γ to S_γ. In [5], PoS is a group of directed acyclic graphs that are "hard-to-pebble". The prover will select of graph of desired size N and store computed labels for each of the N nodes in the graph. The verifier will store a commitment to the graph.
2. *Execution* In the execution phase, the verifier sends a challenge to the prover. The prover solves the challenge using the chosen graph and returns an answer to the verifier. The verifier then checks the answer from the prover and makes sure the prover is storing the nodes from the graph.

PoS is specified by [5] as a tuple of 4 algorithms: $\{Init, Chal, Ans, Vrfy\}$. $Init$ is to initialize the space, $Chal$ creates the challenges, Ans computes the answer to the challenge, and $Vrfy$ verifies that the answer is correct.

In order to understand why a "hard-to-pebble" graph is needed, graph pebbling needs to be explained. Graph pebbling requires using a graph where every vertex is assigned a value. Pebbling a graph can be thought of as determining the value that should be assigned to a vertex v. However, that can only be done if you know the values assigned to each of the vertices on incoming edges of v. A "hard-to-pebble" graph is, therefore, a graph where most of its vertices' values cannot be determined without already knowing a lot of information about the graph. This is of key importance for PoS commitments because it should be required of members to store a substantial amount of data instead of being able to compute needed values without doing so.

2.1. Benefits of Proof of Space

By choosing to invest *disk space* rather than computational power, cryptocurrencies based on PoS have several advantages over cryptocurrency based on PoW.

2.1.1 Ecological

Once disk space is allocated, the cost of computation and accessing the disk will be cheap. In order for PoS to be advantageous over PoW, the complexity and cost of interaction between the prover and verifier must be very small [5]. If they become expensive, there's no advantage to using PoS. A PoS protocol needs to be designed to where all 4 algorithms have low cost.

2.1.2 Economical

As stated earlier, a key assumption of PoS is that users already have unused disk space available, which makes PoS essentially free [5]. More miners will be willing to participate in spite of low reward. In contrast, many miners may not participate or participate in a limited capacity due to cost of expensive computation required by PoW. The reward would have to cover the cost.

2.1.3 Egalitarian

Bitcoin mining is done mainly on large-scale "mining farms" [8] because it is difficult for the average miner to acquire the specialized hardware necessary for mining. PoS blockchain networks are less susceptible to specialized hardware than networks such as Bitcoin. This is because they only require doing a few data lookups every few minutes [8], which is doable even on the slowest of storage hardware.

Spacemint specifically also provides more opportunity to smaller miners by evaluating a miner's proofs based on quality and not simply disk space. This will be explained in detail in . ***Quality of Proofs and Chains***.

2.2. Problems with Proof of Space

Several non-trivial complications arise as a result of the design choices of PoS.

Compared to PoW schemes, PoS schemes require more interaction to work in a blockchain setting. Whereas the interactions in PoW are few and simple, and use a public-coin protocol [8], adapting PoS to a blockchain setting requires some clever solutions.

A major issue has to do with the fundamental problem of fairly determining the winner. Other issues include the "nothing-at-stake" problem and the problem of "challenge grinding". [8]

2.2.1 Determining the winner

Determining a winner becomes problematic when computation power and timing are not inherently available as a way to discriminate between submitted proofs. Whereas timing is used to determine a winner in PoW blockchains such as that of Bitcoin, timing is not an inherent part of PoS based blockchains and leads to winners being picked "unfairly". The aim of PoS is to use disk space as a determiner instead of computation power, so falling back to picking a winner based on speed contradicts this design. One solution to this issue would be to pick winners based on an algorithm that would give every prover a probability of winning that is proportional to however much space it has allocated.

2.2.2 Nothing-at-stake problem

The "nothing-at-stake" problem arises because there is little cost to calculating and submitting a proof once a user has dedicated some disk space to the service. Therefore, a prover could create multiple different blocks with little extra cost and submit the one most advantageous to them.

- *Block grinding* Provers are able to easily try multiple blocks with slight changes each time and little extra computational cost. One solution to this problem is to use an algorithm that restricts proofs to a single unique possibility. [8]

- *Mining on multiple chains* Provers are not incentivized to create blocks for a single "best" chain. [8] Instead, they can easily create proofs for many different chains and submit the proof that would give them the best chance of being accepted. This greatly slows down consensus because it breaks the general agreement among members regarding which chain should be built upon. Resolving this issue requires disincentivizing users from building on blocks and chains that are too old. [8]

3. SpaceMint: How it Works

3.1. Mining

Mining consists of two parts: initialization and mining.

3.1.1 Initialization

Initialization is needed in order to join the SpaceMint network. The miner will specify the N bits of space that he or she wants to allocate for mining. The miner will then generate a (pk, sk) pair, and run the Space commit algorithm with input (pk, N) as a prover.

```
Input: $(\mu, n)$
The prover chooses a hard-to-pebble graph $G$ with $n$
  nodes. The prover then generates a unique nonce $\mu$
  (if this is for initialization, $\mu$ will be the $pk$). The
  prover will label $G$ such that
  $l_i := hash(\mu, i, l_{p1}, ..., l_{p2})$ where $l_i$ is a label for
  vertex $i \in V$ and $l_{p1}, ..., l_{p2}$ are the parent labels of
  $i$. The set that contains all the labels is called $S_\gamma$
  and $\gamma$ is a Merkel Tree with all the $n$ labels.
  Finally, the prover sends $(\mu, \gamma)$ to the verifier.
```

Algorithm 1: Space Commit

In initialization, the prover will be announcing its space commitment (pk, γ) (or in the general case (μ, γ)) through a special transaction, and will be able to mine once this transaction is on the blockchain.

3.1.2 Mining

Once initialization is complete, a miner tries to add a block to the longest chain every time. Here is the pipeline a miner tries:

1. Retrieve the following: hash value of the last block in the best chain, a challenge c (derivation of c will be explained in **. *Where to get the Challenge***). Derive 2 random strings, $\$_p$ and $\$_v$, using c as the seed.

2. Compute the challenges, $(c_1, ..., c_{k_p})$, using $Chal(n, k_p, \$_p)$. k_p is a constant value to ensure that all miners get the same number of challenges.

3. From the challenges, compute a proof of space $a = \{a_1, ..., a_{k_p}\}$ using the following algorithm.

```
Input: $((c_1, ..., c_{k_p}))$
The verifier holds the commitment $\gamma$ and the nonce
  $\mu$. The prover stores $S_\gamma$ and $\mu$.
The prover generates a proof and sends it to the
  verifier: $\{a_i = Ans(\mu, S_\gamma, c_i)\}$ where $i \in k_p$
The verifier checks the proof executing:
  $Vrfy(\mu, \gamma, c_i, a_i)$. The verifier is opening some of
  the committed labels to prove they are stored.
```

   Algorithm 2: Prove Space

4. Compute the quality of the proof: $Quality(pk, \gamma, c, a)$. The quality function is explained in **. *Quality of Proofs and Chains***.

5. If the quality of the proof is high enough and there is a reasonable chance of being the best proof in period i, then compute a proof of correct commitment, $b = \{b_1, ..., b_{k_v}\}$, using the following algorithm. k_v is a value proportional the disk space the miner has allocated.

```
Input: $((c_1, ..., c_{k_v}))$
The verifier holds the commitment $\gamma$ and the nonce
  $\mu$. The prover stores $S_\gamma$ and $\mu$. The challenges are
  generated using $Chal(n, k_v, \$_v)$.
The prover generates a proof for all the labels of the
  nodes and all their parents. This is sent to the
  verifier: $a = Ans(\mu, S_\gamma, c)$
Using $Vrfy(\mu, \gamma, c, a)$, the verifier checks the
  correctness of the proof. The verifier then checks if
  every label, $l_{c_i}$, has been computed correctly in
  Algorithm 1.
```

   Algorithm 3: Prove Commit

   Then create a block and send it to the network. The block will contain proofs a and b and a set of transactions. Details on blockchain structure will be explained in **. *Blockchain Format***. Algorithm 3 runs on the assumption that the prover has properly committed to space in Algorithm 1. The goal of Algorithm 3 is to verify the commitment to space in Algorithm 1 when a miner wants to add a block to the chain.

3.2. Blockchain Format

Similar to Bitcoin, SpaceMint's blockchain is a public account of all transactions. Every block β_i consists of three parts, or sub-blocks, where i is the index of the block. These are

- *Proof chain* sub-block, which contains
  - the current i
  - the miner's signature on the $i-1$ proof chain
  - the space proof, which has the current miner's pk
- *Signature Chain* sub-block, which contains
  - the current i
  - the current miner's signature on the transaction sub-block
  - the miner's signature on the $i-1$ signature chain
- *Transaction* sub-block
  - the current i
  - a list of transactions

From this formalization, it can be seen that in each block, the signature chain connects to the previous signature chain and the proof chain connects to the previous proof chain. It should also be noted that in each block, the signature chain and transaction sub-block are connected, but the proof chain is not. The reason for this is explained in ***Nothing-at-stake problem***.

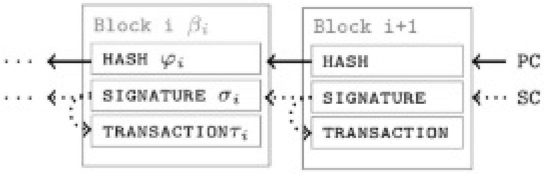

Figure 1: Spacemint Blockchain Structure [8]

3.3. Transactions

SpaceMint has three types of transactions: *space commitments*, *payments*, and *penalties*.

- *Space commitments*

  A space commitment transaction is made by new members joining the network committing with a specific amount of disk space.

- *Payments*

  A payment transaction consists of a list of input coins and a list of output beneficiaries [8]. Each user contributing to the list of input coins must also include in the transaction a signature of the entire output list, verifying their consent to give their coin.

- *Penalties*

  Penalties are used to punish miners for engaging in malicious behavior. A penalty transaction will consist of a proof of malicious behavior submitted by one miner accusing another. The primary, but not only, use of penalties is to dissuade miners from mining on multiple different chains [8]. To punish such an action, the proof can contain two blocks with the same index signed by the same miner.

3.4. Where to get the Challenge

In Bitcoin, which uses a PoW scheme, it gets its challenge for the current block i by using the hash of block $i-1$. Unfortunately, when using a PoS scheme, obtaining its challenge from the $i-1$ block could actually hinder consensus. If there were multiple chains and the challenge came from the $i-1$ block, then any rational miner would try to find proofs for as many chains as possible. This is because in PoS, proofs can be found quickly, and by finding proofs for multiple chains, you are more likely to get a proof that has a probability of having a high quality.

As a result, the challenge for β_i comes from the $\beta_{i-\triangle}$ block. In [8], they suggest $\triangle = 50$, as they believe that this is a reasonably large value such that the probability of having multiple chains surviving is unlikely. The same challenge will be used for δ consecutive blocks, and [8] suggests $\delta = 10$. The challenge will also be only from the proof chain, rather than both the proof chain and the signature chain. These choices are explained in ***Nothing-at-stake problem***.

3.5. Quality of Proofs and Chains

3.5.1 Quality of a Proof

In order to fairly pick a winner from among multiple correct submitted proofs, SpaceMint calculates a "quality" for each proof. The quality of a submitted proof is calculated by the hash of the proof itself. The proof which results in the highest quality is the one that is accepted. Furthermore, SpaceMint works such that the probability of a proof being the highest quality is proportional to the percentage of the total committed space that the prover holds.

This is done by first normalizing the hash of the proof (based on the commitment size of the prover) to fall within the range $[0, 1]$, and then raising the value to the power of $1/N$ (where N is the total space committed in SpaceMint).

This modifies the probability distribution to be based on the proportion of committed size.

3.5.2 Quality of a Chain

Having a good notion for the quality of a chain is also important. Calculating the quality of a particular chain is an extension of calculating the quality of each block in the chain. Specifically, it is the product of the blocks (proofs). One additional term needed is an increasing multiplier applied for more recent blocks (needed in practice because quality of older blocks is degraded compared to newer blocks).

3.6. Solutions to Proof of Space issues

SpaceMint is one of the few real-world and practical implementations of a PoS blockchain network. Because it aims to be usable in practice, it must provide an answer to the problems that arise from a naïve PoS implementation. Thus it has come up with several solutions to previously mentioned obstacles.

In order to adapt a PoS scheme to a blockchain setting, SpaceMint uses the Fiat-Shamir paradigm [8]. This solves the problem of *interactivity*, that the original definition of PoS suffered from as an interactive protocol. Instead of using an interactive public-coin challenge, SpaceMint's use of the Fiat-Shamir heuristic allows its proofs to be constructed without interaction, using a hash of the previous message. This increases efficiency and is already in use in other blockchain environments like Bitcoin. Likewise, for bookkeeping reasons, SpaceMint uses the blockchain infrastructure itself to record certain messages of the protocol.

In addition, the following solutions help to readjust the user incentives in order to create a network that is less prone to abuse.

3.6.1 Determining the winner

SpaceMint uses its calculation of the quality of a proof in order to fairly determine a winner from among multiple submitted proofs. Most importantly, the probability of any particular prover winning out is indeed proportional to the amount of space they have dedicated to SpaceMint. [8] An added benefit of SpaceMint's quality algorithm is that, since the quality of a proof is deterministic, a prover can calculate for themselves the probability that their proof will be accepted [6].

3.6.2 Nothing-at-stake problem

- *Block grinding*

  SpaceMint's solution to block grinding is to decouple proofs from transactions and create two separate chains [8]. In particular proofs depend only on the *proof chain*, limiting the possibilities to a single unique proof. This avoids the chance of block grinding by slightly modifying the block.

- *Mining on multiple chains*

  SpaceMint tries to diminish the consensus issues arising from miners submitting blocks for many different chains by requiring proofs to include not the hash of the last block but the hash of the nth-to-last block. [8] This aids consensus by decreasing the time that forks of more than n blocks survive.

4. Evaluation

This evaluation is based on a prototype of Spacemint, implemented by [8]. The implementation was done using the programming language Go and the hash function used was SHA3 in 256-bit mode for its hash function. The experiments were done on an Intel CPU with 8 GB of memory accompanied by a 2-TB disk drive with 64 MB of cache.

4.1. Initialization Time

The first step needed in Spacemint mining is the initialization of space that is done in Algorithm 1. Hash values need to be computed for all nodes and a Merkle tree must be computed over the hashes. The results from [8] show that computation time grows linearly with the quantity of space initialized. Initializing around a terabyte of space can take about 24 hours, however, this process is only done once as the space will be used repeatedly. Investing a non-trivial amount of time in initialization prevents miners from using the *same* space for different commitments.

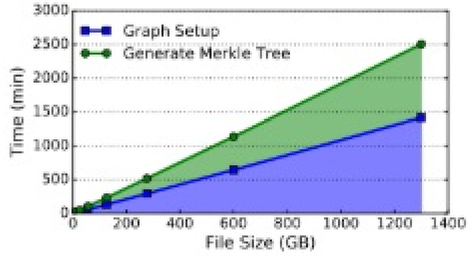

Figure 2: Initialization Time [8]

4.2. Proof Size

Spacemint proofs are generated in Algorithms 2 and 3. We need to compute an answer for each of the $\lambda * log(n) + 1$ nodes, where λ is a security parameter. In Algorithm 3, $k_v = \lambda * log(n) + 1$ and $k_p << k_v$. The answer for each node is of size $log(n) * 32$ bytes and each node has at most 2 parents. This makes the overall proof size upper-bounded at $3 * \lambda * log^2(n) * 32$ bytes. The figure below shows the size of proof when $\lambda = 30$.

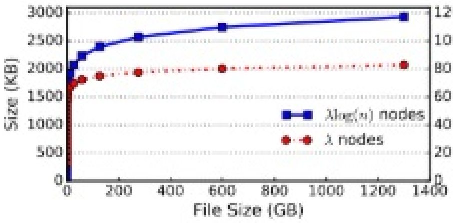

Figure 3: Proof Size [8]

4.3. Generation & Verification Time

In Algorithm 2, when verifying the proof sent by the prover, the verifier opens k_p nodes in the Merkle tree tt check the solution. It takes less than 1 millisecond to read a hash from the disk, making the entire verification take only a fraction of a second.

If the proof is of good enough quality, then Algorithm 3 will be run where a larger proof is generated. In Algorithm 3, proofs must be generated not just for each node, but for the parents of the nodes. This process takes at most 30 seconds, most times closer to 20.

The generation of proofs in Spacemint takes longer than Bitcoin and requires multiple hash calculations to verify. However, as the graph below shows, the verification time and size of new proofs added is extremely marginal compared to the transactions added with every block.

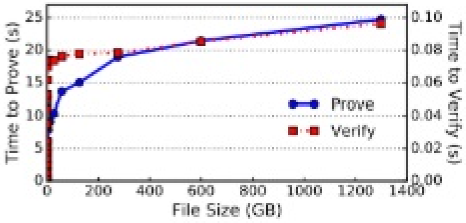

Figure 4: Generation & Verification Time [8]

4.4. Energy Estimates

The prototype used by [8] was experimented with on a full CPU, which wastes a lot of computing power. Most miners would mine on more effiecient devices that consume less the 10 Watts of energy. Most miners will only run a few nodes at a time and of all proofs generated by miners, only a very small percentage will be considered good enough to make it onto the blockchain. To get an upper limit on the energy cost, consider a scenario where there are 100,000 miners, each with 1 TB of space, and 1% of the miners mine answers good enough to get onto the blockchain (1% is a conservative estimate). The equations is as follows:
$Power * \#ofminers * VerificationTime +$
$Power * GoodMiners * GenerationTime =$
$10W * 100000 * 0.01 + 10W * 1000 * 20 = 210$ kJ/block
This translates to 210 kJ per minute. Bitcoin on the other hand, consumes about 6 GJ per minute, many magnitudes greater than Spacemint.

4.5. Storage Medium

Bitcoin mining has been dominated by ASICs, specialized hardware that can compute hashes a lot faster than a regular processor can that people have on their computer. While [8] suspects that SpaceMint would not run into specialized hardware issues, there is not enough real-life use of PoS to say for sure. They argue that there are cheaper alternatives to the storage medium compared to hard disks like tape, but they believe that tape would not be a good use to decrease the cost of storage because mining requires frequent random access, which tapes are very slow at. They also point out that there are more expensive alternatives such as solid state drives which makes random access a lot faster, but these benefits are not substantial as mining does not require enough random lookups for the benefits of having a solid state drive outweigh the cost of a hard disk. As a result, hard disks seem to be the best suitable storage as it is relatively cheap, and the lookup is not too slow. Hard disks are also general purposed, and thus, should be readily available to most people.

5. Conclusions

Blockchains that rely on PoS commitments appear to successfully solve the inequality issues and the cost-prohibitive nature of blockchains that use PoW. Certainly, giving up spare disk space is much more user-friendly than needing to buy computer hardware and being required to continually use energy to run the computations. However, practical implementation of PoS-based blockchain networks must address the issues that a naïve implementation would be plagued with. SpaceMint is a credible contender for being a practical implementation that sets up the correct incentives for its users and achieves the goals of PoS in its cost-effectiveness.

5.1. Benefits of SpaceMint

The game theoretic properties of SpaceMint are conducive to a practical and real-world usable blockchain. In particular, SpaceMint as a *game* has an equilibrium that incentivizes members to stick to the protocol [8]. Because of this, SpaceMint proves to be a practical blockchain network that has the advantage over PoW-based blockchains that it does not require immense computational power while still remaining fair.

We have already seen that SpaceMint is more ecologically sustainable, in terms of smaller energy requirements, and more economical in terms of start-up costs for a prospective member. SpaceMint is a cryptocurrency that is friendly to "small users" in regards to the investment costs and energy costs of mining. This discourages the "mining oligarchy" of cryptocurrencies such as Bitcoin and leads to a more even distribution of commitments because mining power is not affected as much by specialized hardware [8] that PoW-based blockchains are affected by, such as ASICs. Furthermore, it still treats each user fairly by having the probability of a mined block being accepted be proportional to the size of their commitment.

5.2. Issues with SpaceMint and Proof of Space

The hurdles of implementing SpaceMint include the necessity of synchronization of block creation time steps. [4] This is needed because PoS is inherently not a time-based protocol unlike PoW. Whereas PoW schemes can depend on the time it takes to create blocks to create a good rate for the progression of the blockchain, PoS schemes take a trivial amount of time to compute. Thus, a separate synchronization must be done to ensure the stability of the blockchain.

A potential issue regarding PoS is its design potentially making 51% attacks easier to perform than on a coin that uses the PoW mechanism. As no specialized hardware is required and hard disks are fairly cheap, an adversary may more easily own 51% of the networks space for small coins. However, this problem is not entirely unique of PoS schemes.

5.3. Viability of Proof of Space and SpaceMint

Using a Proof of Space scheme is not a brand new idea, but only more recently implemented and used in a blockchain setting. SpaceMint is also not the first to describe an implementation.

Burstcoin is a cryptocurrency that uses disk space as its primary mining resource. [8] However, unlike SpaceMint, Burstcoin's PoS implementation is problematic, because miners are able to get away with storing only 10% of the size of the data they have committed to by calculating many values quickly on-demand.

Chia is a cryptocurrency based on Proof of Space. Chia is actually inspired by the PoS of SpaceMint. Chia builds upon the the quality calculation introduced in [8] by combining it with a verifiable delay function (VDF) in order to ensure that the highest quality proof is actually the proof that is able to be calculated the fastest. [4] This is done to avoid the need for synchronization of the blockchain like SpaceMint does. The downside is that this modification moves away from a purely Proof of Space scheme by including concepts that fall under "Proof of Time". However, even with this departure, Proof of Space is still a requirement within the Chia Network, so it is a credible and practical implementation of Proof of Space in the real world.

Filecoin uses a somewhat related scheme called "Proof of Spacetime" (not to be confused with the combination of Proof of Space and Proof of Time). This requires miners to prove that they have stored a piece of data continually for a certain period of time; this is done by periodically producing sequential proofs of space and composing them recursively to generate the final proof [3]. The aims of Filecoin are different than those of SpaceMint, since it can and is intended to be used to store useful data on the network (made apparent by its name) as a form of cloud storage.

The major draw of Proof of Space over Proof of Work is that mining does not require much energy or investment, and SpaceMint is faithful to this idea. SpaceMint is the only cryptocurrency we can find that is based purely on Proof of Space. Because of the characteristics of its implementation, it is more accessible to small miners, it is conducive to a less centralized network than ones using Proof of Work, and despite certain aspects of its structure being cumbersome, it provides a valid and working concept for Proof of Space.

References

[1] Bitcoin energy consumption index. https://digiconomist.net/bitcoin-energy-consumption.

[2] Hashrate distribution. https://www.blockchain.com/charts/pools.

[3] Filecoin: A decentralized storage network. Technical report, Protocol Labs, 2017. https://filecoin.io/filecoin.pdf.

[4] B. Cohen and K. Pietrzak. The chia network blockchain. Technical report, Chia Network, 2019. https://www.chia.net/assets/ChiaGreenPaper.pdf.

[5] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak. Proofs of space. Cryptology ePrint Archive, Report 2013/796, 2013. https://eprint.iacr.org/2013/796.

[6] T. Moran and I. Orlov. Simple proofs of space-time and rational proofs of storage. Cryptology ePrint Archive, Report 2016/035, 2016. https://eprint.iacr.org/2016/035.

[7] A. Narayanan, J. Bonneau, E. W. Felten, A. Miller, and S. Goldfeder. *Bitcoin and Cryptocurrency Technologies - A Comprehensive Introduction.* Princeton University Press, 2016.

[8] S. Park, A. Kwon, G. Fuchsbauer, P. Gaži, J. Alwen, and K. Pietrzak. Spacemint: A cryptocurrency based on proofs of space. Cryptology ePrint Archive, Report 2015/528, 2015. https://eprint.iacr.org/2015/528.

[9] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, and A. Mohaisen. Exploring the attack surface of blockchain: A systematic overview, 2019.

[10] M. Szmigiera. Number of blockchain wallet users worldwide from 3rd quarter 2016 to 4th quarter 2019. https://www.statista.com/statistics/647374/worldwide-blockchain-wallet-users/, 2020.