Proof of Space with VDF:

An Alternative Permissionless BFT Consensus Protocol

Yuxuan Luo
2021/12/15

1. Background

A permissionless, Byzantine fault tolerant consensus protocol is an essential infrastructure for any blockchain that is public and decentralized. Bitcoin was the first of its kind to utilize Proof of Work along with the longest chain consensus to achieve this goal. PoW provides Bitcoin with the robustness against sybil attacks through the use of a scarce and cryptographically verifiable resource - hashing power[1].

While PoW has proven itself to be a straightforward and yet effective solution of the consensus problem, criticism on its downsides grew as time goes on[2]. Some believe that the environmental impact of PoW is unsustainable. Others fear that the rise of ASIC miners is leading to greater centralization.

Over the years, a number of alternatives to PoW has been proposed, most notably Proof of Stake. PoS replaces PoW's energy intensive computation with tokens locked as the scarce resource. While this approach does address the energy consumption concerns of PoW, it loses in many security and decentralization aspects. Whereas PoW is immune to attacks such as stalling, grinding attacks, long range attacks, and nothing-at-stake attacks due to its "grindy" nature, the same cannot be said about PoS. In order to protect the protocol against aforementioned attacks, PoS consensus protocols had to sacrifice decentralization for security, and rely on (relatively centralized) delegations, predicable leader elections, clock synchronization assumptions, and network assumptions[2][3][4].

In an era where permissionless Proof of Stake is yet to be tested and adopted[4], could there be an uncompromised approach that keeps the best of both worlds?

2. Proof of Space

2.1 PoSpace Concept

What if instead of token staked or hashing power, disk space were to be used as the scarce resource for sybil resistance? How can one design a consensus protocol where someone dedicating α gigabytes of storage to a network of β gigabytes is expected to have a α/β chance of being elected as the leader?

Several obstacles with this approach should become immediately obvious.

a) How does other miners cryptographically verify that a miner actually owns the amount of storage they claim to own?

b) How to sample the miners and select a winner among them?

c) How does one verify the miner is dedicating their disk space to the network and not using it for something else?

d) Since it is computationally cheap to do so, how to stop the miners from generating the proof of storage with multiple public keys to gain more netspace share?

e) Since it is computationally cheap to do so, how to stop grinding attacks and nothing-at-stake attacks?

f) Should miners with faster SSDs be able to generate proofs faster than miners with HDDs?

One implementation known as SpaceMint that addresses these problems was proposed in 2018[5]. The following sections explains how SpaceMint is able to make disk space dedicated to the network a cryptographically verifiable resource.

2.2 SpaceMint Implementation

2.2.1 Crypto Puzzle & the enforced necessity of Storage Dedication

To make sure that a miner has indeed dedicated a chunk of storage space to the protocol, miners are challenged with verifiable crypto puzzles that is impossible to compute at real time. The miners can only reply with the solution if they have previously computed and stored potential answers in a hard drive ready to be looked up.

Suppose a miner wants to dedicated $N \approx \ell \cdot 2 \cdot 2^{\ell}$ bits of storage space to the network, we can define two functions $f(x)$ and $g(x)$.

$$f(x) = H(pk, x)_{|\ell} \quad \text{and} \quad g(x, x') = H(pk, x, x')_{|\ell}$$

Where $X_{|\ell}$ denotes the ℓ-bit prefix of X, H is a random hash function like SHA256, pk is the public key of the farmer, x and x' are just two different random numbers.

The cryptographic puzzle y comes in the form of a bit string of length ℓ, $y \in \{0,1\}^{\ell}$, and the miners are expect to supply the solution tuples

$$(pk, x, x') \quad \text{where} \quad x \neq x \quad \text{but} \quad f(x) = f(x') \quad \text{and} \quad g(x, x') = y$$

To supply the solution without computing *f* and *g* on the fly, the miners compute and store a table with the tuples $(x, f(x))$ sorted by $f(x)$. Then, take all tuples (x, x') where $x \neq x'$ but $f(x) = f(x')$ and compute $g(x, x')$ and store $(x, x', g(x, x'))$ in a table sorted by $g(x, x')$. With the second table, miners can now find, in logarithmic time, all $y = g(x, x')$.

In practice this process is repeated several times with 7 tables to stop Hellman attacks detailed in 'Beyond Hellman'[14].

Some observations can be made here:

- H is assumed to be output uniformly random results and therefore cannot be predicted or reverse engineered. This makes precomputing the hashes and store them on the disk the only way to reliable find answers to the puzzle before the round ends.
- For the same reason, an incentivized miner cannot realistically grind out solutions to challenges as they arrive.
- The more chunks of N bits storage filled with $(x, x', g(x, x'))$, proportionally more proofs to the puzzle the miner will find.
- Because $g(x, x')$ is hashed with the miner's public key, the same block of storage cannot be used to generate proof for another identity, giving PoSpace sybil resistance.
- Because the xs and x's are discrete random variables, their distribution follows the Poisson distribution. As a result, $E(X) = \sum_0^\infty x \frac{e^{-\lambda}\lambda^x}{x!} = \lambda$ suggests that the expected number of proof is 1.

2.2.2 Where do the challenges come from?

If miners can predict what the future challenges will be, they can simply do the computations before hand and only save the $(x, x', g(x, x'))$ tuples that are going to satisfy the challenges. In that case, the mining operation again becomes compute bounded, just like PoW.

To deny this possible circumvention of dishonest miners, the crypto puzzle is derived from the hash of the previous block, meaning if dishonest miners were to grind for a solution, they only have the time from when the wining block of the previous height is determined, until the current round ends.

2.2.3 How is the winner decided?

There is only one ungrindable crypto puzzle per height, no concept of difficulty, and each block of *N* bits worth of hashes is expected to find one proof. Then how is the winner decided?

Clearly the winner can't be the first one to find such a proof. Since the time it takes to find the proof is very short compare to network delays, such winner selection protocol will lead an abysmal chain growth.

Instead, we want to elect a miner as the leader with a probability equals to their share of the total network space. SpaceMint achieve this by calculating a Quality score of best proof each miner has for a given puzzle. A function for calculating the Quality score which also satisfies the fair share property is

$$D_{N_{\gamma_i}}(\mathsf{hash}(a_i)) := \left(\mathsf{hash}(a_i)/2^L\right)^{1/N}.$$

The proof is left as an exercise for the reader.

In essence, the Quality of a proof is the hash of the proof normalized to the range [0,1], with 0 considered the worst and 1 considered the highest quality. A miner who dedicate more N-bits block to store the hashes is expected to find proportionally more proofs. Furthermore, since the Quality scores are hashes, they are randomly distributed. Miners with more proofs get more attempts at the Quality hash function, and are therefore given a better chance at finding the highest Quality score for each crypto puzzle.

2.3 SpaceMint Safety and Liveness Guarantees

After taking a look at a simplified version of SpaceMint. It can be said that SpaceMint is good against grinding attacks (since input to the hash function is pre-calculated), and sybil attack (miner's public key is used to compute the hashes store on disk).

At this point, it is unclear how proof of space can stop 51% attacks, nothing-at-stake attacks, long range attacks, or selfish mining. SpaceMint slashes the miner's staked token to prevent adversarial behavior and to enforce a block time based on clocks.

Even though SpaceMint's PoSpace keeps some of the security guarantees of PoW, it still leaves much to be desired. Can we do better?

3. Verifiable Delay Function

3.1 Properties of a VDF

A VDF is a function of three algorithms, `Setup`, `Eval` and `Verify`, that meets following requirements[6]:

- *Sequential*: honest parties can compute the output and a proof in *t* sequential steps. However, adversaries with more parallel computing power cannot compute the output or proof in significantly fewer steps
- *Efficiently verifiable*: the output can be efficiently verified with the proof.
- *Unique*: for all inputs, it is computationally hard to find an alternative output to `Eval` that will be accepted by the `Verify` function.

More formally, `Eval` takes a challenge *c* and a time parameter *t* and outputs a proof

$$\tau = (y, \pi)$$

Here y is the output and π is a proof to efficiently verify that y has been computed correctly.

`Verify` either accepts or reject (y, π) but it always accepts a valid τ.

$$\forall t, c : \text{Verify}(\text{Eval}(c, t)) = \text{accept}$$

Moreover, it is computationally difficult to find a *y* such that `Verify`(y, π) = accept and `Eval`$(c, t) \neq y$.

3.2 Observations

VDFs can be quite useful for a non-PoW consensus protocol as a time keeper. Protocol like Tendermint that requires vote counting needs to figure out ways to enforce the monotonicity and validity of timestamps. VDFs provides a cryptographically verifiable way to mark the passage of a period of time. Its hard to compute, easy to verify property means that any miner cannot advance the round faster with more parallel computation resources, and that validators can quickly verify the submitter of the VDF proof actually took the time to compute it.

It also means that as long as more than one person is generating VDF proofs in the network, the consensus can progress.

3.3 Example of a VDF

One example of a VDF, that is used by Chia, is the squaring modulus function[7].

Let the challenge candidates group be $c \in \mathbb{Z}_N^*$ where N is the product of two large primes. `Eval`(c,t) returns $(y = c^{2^t} \bmod N, \pi)$. Here y is computed by squaring *c t* times sequentially $c \to c^2 \to c^{2^2} \to \cdots \to c^{2^t}$. It is conjectured that there is no shortcut to

compute *y* without doing integer factorization on N. Since integer factorization itself is a NP Hard problem, the squaring modulus function can be considered *Sequential* and as a result of that *unique.*

According to source [7] and [8], with π, the verifier only needs to perform $\log_2 t$ computations to verify the output *y,* which makes this function also *efficiently verifiable.* Chia uses the Wesolowski method to verify the VDF proof, of which the prover can compute the proof efficiently using a prime number sent by the verifier. The verify only needs one round of communication with the prover (As opposed to Pietrzak's method that requires $\log_2(T)$ rounds of back and forth communication in a divide and conquer style verification). Unfortunately I'm to dumb to understand why the Wesolowksi method works, so I'll just link his work here [10]

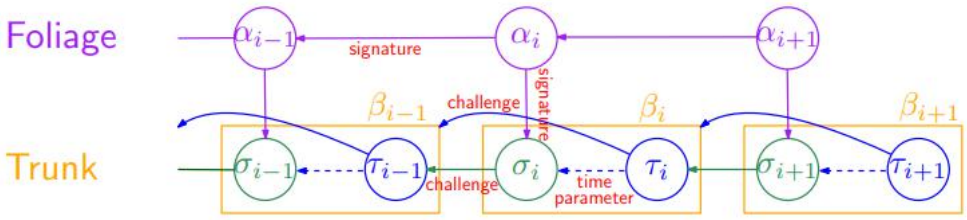
4. Chia - SpaceMint with VDF

This section references Chia's pre-mainnet launch green paper, which is quite different from the current mainnet implementation.

4.1 The role of VDF in Chia's Consensus Model

Previously we saw the flaws in SpaceMint's design that had to be remedied by staking and slashing. Chia introduces another type of participants in the consensus layer called Timelords, whose job is to infuse the blocks with a VDF proof to finalize them[9].

Just like SpaceMint, Chia's blockchain is split into two chains, the foliage chain and the trunk chain.



Each block in the foliage chain contains a signed pointer to the previous block, a signed pointer to a proof block in the trunk chain at the same height, and transaction data. The foliage chain is grind-able with different transaction data, but doing so doesn't give the miners any advantage as neither σ_i nor τ_i is dependent on α_i.

Each block in the trunk chain contains σ_i, the proof of space whose crypto puzzle is the VDF proof of the previous block, and τ_i, the VDF proof whose input is the hash of the previous block and the $0.H(\sigma_{i-1}) \cdot T$ the difficulty parameter.

4.2 Block time

This design implicitly solves SpaceMint's lack of a flexible adjustable block time. Block miners cannot proceed to mine the next block until a Timelord generates a VDF proof to finalize the current block. Timelords are also government by the difficulty parameter T to ensure that block time stays relatively stable even when the Timelords have gotten faster at computing the outputs and proofs of the VDF.

4.3 Long range attack

Long range attack can be a problem for naive PoSpace implementations. It is not a threat to PoW where the longest chain is determined by the total number of expected hashes of the chain. Even if an adversary manages to get over 50% of all hash power, they still need to accumulate the difficulty of its private chain before it can reach the difficulty of the current longest chain. By that time, the honest longest chain will be further ahead in difficulty, and it will still take the adversary a long time to catch up.

However, in a PoSpace context, because looking up the proof is computationally very cheap, an adversary, if without obstruction, can grow a fork in a very short amount of time[9]. Consider an adversary with a reasonable amount of capacity storing proofs compare to the that of the network at one point, they can potentially provide the highest quality proof for some early challenges and start their own fork of the blockchain.

SpaceMint's proposal prevents long range attack by putting more weight on the more recent blocks. On Chia's blockchain, such attack is prevented by the requirement of VDF proofs generation. The adversaries wishing to start long range attacks on Chia need to recompute the VDF proofs for their own blocks, which is limited by the rate of their ability compute sequential squaring modulus. Hence, VDF protects Chia from long range attack like PoW's protection. A minor difference is that the adversary doing a long range attack can still mine as an honest miner since the VDF proof for the to head they are extending and for the current head of the longest chain are different and can thus be computed in parallel.

4.4 Grinding Attacks

Since the foliage blocks are grind-able, adversaries may attempt to create multiple timestamps for the last block before difficulty adjustment, so that they can get different difficulty inputs to the VDF for the next block and then compute them in parallel. This attack gives the adversary different VDF outputs for the blocks at the first height of the new epoch. Since different VDF proofs changes the cryptopuzzle for the next space proof, the adversary can now grind on the subsequent height until they find a proof with a high Quality score.

Chia counters this type of attack by shifting the period to compute the new difficulty one fourth of an epoch backwards. This alternation means that for the adversary to take advantage of that one VDF proof, they will have to extend their chains for a quarter of the epoch to realize their gains while keeping their chain's Quality score above the that of the public chain.

4.5 Nothing at Stake Attacks

Nothing at stake attacks refers to the action of adversaries attempting to extend all recent blocks of a longest chain consensus PoStake or PoSpace system. This is a real problem for PoS and PoSpace blockchains where computing the proof is cheap, and by doing so the adversaries makes the mining process more grind-able.

Chia uses variety of unique solutions to counter this type of attacks. It employs a concept called sub-epoch of 32 blocks to limit the number of blocks an attacker can try to extend at a time. The goal achieved by this tactic is that "challenges aren't discovered until they need to be responded to, and the source of challenges is already thoroughly buried or orphaned by the time its results are discovered"[15]

Chia calculated that an adversary would need over 46% of the total net space to consistently pull off a double spend attack. This is helped by the fact that Chia employs a concept called sub-epoch where every 32 blocks share the same challenge, and the attacker can only try about 32 different combinations, giving it a small boost in effective netspace.

Bram Cohen told me on Reddit that this part is poorly explained so I blame my inability to understand this stuff on him...

4.6 Liveness

The Green Paper states that an adversary who cannot break the security of the signature scheme cannot slow down the rate at which the honest farmers grow the chain. This combined with the fact that Chia is permissionless and doesn't require vote counting, it can be said that Chia has the same liveness guarantee as a PoW system such as any other longest chain consensus (if there is a k-balanced leader sequence then transactions will eventually end up on the longest chain).

4.7 Problems with VDF

While the previous assumption that there is no short cuts to compute squaring modulus, there can certainly be minor improvements in software or hardware which makes VDF generation faster. Having a private and faster Timelord will allow an adversary to add blocks to the blockchain at a higher rate than waiting for a public Timelord to announce the VDF for the current height. This will, in effect, nerf the effective capacity honest miners by nerfing the public Timelord, after the difficulty adjustment kicks in.

When the adversary colludes with a mining pool, they could potentially attract incentivized miners to join and use the combined net space for a 51% attack.

4.8 Summary

The introduction of VDF to PoSpace allows Chia to be truly permissionless, in the sense that anyone can join and leave at anytime, and do whatever they wish for as long as they want, which is not previously possible under SpaceMint. In [3], Chia network argued that it is unlikely for someone to come up with a Timelord fast than anything publicly available for a competitive advantage, and the network remains secure as long as one of these public Timelords is running. However, I would still consider this as a security trade off for giving up PoW.

5. Energy Saving with PoSpace

5.1 Background

Just like the ASIC miners and GPUs used for PoW, hard drives used for PoSpace also outputs carbon dioxide during the manufacturing process and take electricity to run. Therefore, PoSpace still carries a significant environmental impact compared to PoS. However, compare to PoW, PoSpace looks much greener on paper. One might find it interesting to see whether PoSpace is actually green or is just proof of electricity with extra steps.

I will be referencing this table throughout this section. The most energy efficient mining hardware is selected from the Bitcoin camp, the Ethereum camp, and the Chia camp.

Mining Equipment	Hash Rate	Power Consumption (W)	Market Price ($)
Antminer S19 Pro	110 TH/s	3250	15000
3060 Ti	60.6 MH/s	125	900
18TB HDD	18 TB?	5	360

5.2 The Arguments for PoSpace

Running a hard drive consumes far less electricity than running an ASIC or GPU. However, if mining for a PoSpace blockchain is profitable, miner will still buy loads of hard drives, which arguably becomes e-waste. Chia makes an interesting prediction that because the main cost of running a Chia PoSpace miner is the up front hardware cost, the miner profit equilibrium will result in negative profitability if buying new hardware.

In class, we discussed that the network difficulty ↔ miner probability relation will converge to an equilibrium where buying new mining hardware is just profitable enough for some to do it. Chia argues that due to the low ratio between electricity cost and hardware cost (10% of Ethereum, and 6% of Bitcoin), game theory would suggest that miners seek out a competitive edge by using underutilized storage or second hand storage to optimize the up front hardware cost.

If this prediction is true, then we can say that if the same amount of capital engages in PoW and PoSpace mining, PoSpace mining will generate less e-waste (measured by $) and consume less energy in the mean time.

5.3 Associated Cost With PoSpace

As mentioned in Section 2, for PoSpace to work, the hard drive needs to be filled with some hashes first, and generating and sorting these hashes takes additional power. In our case, a 300W Ryzen 5950X system generates a 101GiB sized sorted tables of hashes in 20 minutes, implying that filling each hard drive costs around 18 kilowatt hour of electricity, equivalent to the energy used to run the hard drive for 150 days. In

the long run, this pre-processing energy cost is just a constant, and will be dismissed for the calculations below.

There is also additional energy consumption of the system that performs the crypto puzzle searches. People have been using Raspberry Pi's and USB expanders to connect the hard drives. This setup incur little additional energy consumption. We've been using servers each with a Xeon E3 processor and 12 hard drive bays to host the hard drives. In practice, we observed 30W of idle power and 90W during mining. So even with our not so efficient setup, the host PC is only a small constant factor to the power consumption of the hard drive. Therefore, the energy cost to run the PC is omitted in the following calculations as well.

5.4 Current Network Statistics

Blockchain	**Network Hash Rate / Capacity**	**Market Cap (Billion USD)**	**24hr Volume (Billion USD)**
Chia	31590000 TeraByte	0.245435	0.017279
Bitcoin	169 588 000 TH/s	918.671	24.636
Ethereum	924.27 TH/s	479.596	22.651

* The network stats from the three blockchains in question at the time of writing. Chia's Market Cap does not take its premine into account.

In this section, I will attempt to compare the energy consumption of the three Blockchains with respect to the the values they produce(in Market capitalization and 24-hour trade volume). The calculations will be done with the energy consumption estimate done by Digiconomist and Chia [11] [12] [13], as well as with a theoretical lower bound when only the most energy efficient mining hardware is used.

5.5 PoSpace Energy Efficiency

Blockchain	Energy Consumption Estimate (TWh per year)	Market Cap Per Energy Consumption (Billion$ / TWh/ year)	24hr Volume Per Energy Consumption (Billion$ / TWh/ year)	MC Per EC Relative to Chia	24hrV Per EC Relative to Chia
Chia	0.22482	1.0917	0.07685	1	1
Bitcoin	201.81	4.5521	0.1220	4.169	1.5875
Ethereum	98.56	4.8660	0.2298	4.457	2.990

* Done using estimates from [11] [12] [13]

Blockchain	Energy Consumption Estimate (TWh per year)	Market Cap Per Energy Consumption	24hr Volume Per Energy Consumption	MC Per EC Relative to Chia	24hrV Per EC Relative to Chia
Chia	0.07686	3.193	0.2248	1	1
Bitcoin	43.89	20.93	0.5613	6.554	2.496
Ethereum	16.70	28.71	1.356	8.991	6.032

* Done using most efficient miners for each blockchain

If we were to quantify the value produce by a blockchain by its market cap and trading volume, then Chia, in its current state, is even less environmentally friendly than Bitcoin and Ethereum, the two most power hungry PoW blockchains.

However, there is another way to quantify the value of Proof of Space, which is by the security it provides per electricity consumed. For any PoW longest chain blockchain, the threshold that the adversary needs for a double spend attack is 51% of the overall hash rate, whereas the threshold for a Chia-like PoSpace longest chain consensus blockchain is 46%.

Blockchain	Energy Consumption (Watt/hr) per $ Spent on Mining Equipment	Effectiveness of $ Spent on Mining Equipment Against Double Spend	Effective Energy Consumption (Watt/hr) for Security per $ Spent on Mining Equipment	EEC for Security per $PoME Relative to Chia
Chia	0.01388	0.46	0.006384	1
Bitcoin	0.2166	0.51	0.1104	0.05779
Ethereum	0.1388	0.51	0.07078	0.09019

* Done using most efficient miners for each blockchain

By this metric, electricity used by Chia is 17 times more efficient than used by Bitcoin and 11 times more effected than used by Ethereum.

6. Smart Contract on Chia

This section was originally planned for a look into Chia's UTXO smart contract implementation known as smart coins. Unfortunately I never found any teammate to help me out... so this section has been left short.

In a nutshell, the Chia unit coins are bundled into a *spend bundle,* which is identified by the hash of the string describing them, and sent to full nodes[15]. The spend bundle includes a spending condition in the form of a puzzle hash. The (Turing complete) spending conditions can be nested, and can be used to enforce certain conditions be satisfied for the coin to be spendable or specify what would happen after the coin is spent (minting new coins etc). This allows the users to put their own rules to their coins and gives them stronger control. Applications can also be designed with free participation or sign up restrictions as users as users simply create a coin that follows the rule set.

The UTXO model allows Chia's smart contract to be evaluated in parallel, making it easier to scale. It also consumes less resource compared to the Ethereum model since it doesn't need to store and update the state. The smart coin model also means that MEV is less of an issue for Chia due to the decentalized nature of the local smart contract.

However, a UTXO can only be spent once, which complicates some smart contracts.

7. Everything is a Race and Nakamoto Still Winning

To conclude, PoSpace is an alternative to PoW developed to use storage capacity as the scarce resource against sybil attacks in consensus. SpaceMint proposed a solution that still required staking and slashing of tokens to punish adversarial behaviors. Chia integrates VDF into the formula to fix some of the vulnerabilities of PoSpace and made it permissionless. Still, the security guarantees of Chia isn't as strong as a PoW consensus protocol.

On the power consumption front, a PoSpace system in theory can consume less energy and produce less e-waste than PoW systems. Chia's implementation of PoSpace also suggests that it is a lot more efficient in terms of energy usage for security. But at the moment, the market cap or trade volume of Chia isn't justifying this advantage for the electricity it uses.

[1] https://bitcoinwhitepaper.co/bitcoin.pdf

[2] https://web.stanford.edu/class/ee374/downloads/notes14.pdf

[3] https://www.chia.net/assets/Chia-New-Consensus-0.9.pdf

[4] http://tselab.stanford.edu/downloads/PoS_LC_SBC2020.pdf

[5] https://sunoopark.com/p/18/spacemint.pdf

[6] https://eprint.iacr.org/2018/601.pdf

[7] https://theory.stanford.edu/~dabo/papers/VDFsurvey.pdf

[8] https://github.com/Chia-Network/chiavdf/blob/0cd64249119b9c1d68e18a0fcfb76a4dcc01678d/src/verifier.h

[9] https://www.chia.net/greenpaper

[10] https://eprint.iacr.org/2018/623

[11] https://chiapower.org/

[12] https://digiconomist.net/bitcoin-energy-consumption

[13] https://digiconomist.net/ethereum-energy-consumption

[14] https://eprint.iacr.org/2017/893.pdf

[15] https://www.reddit.com/r/chia/comments/ri8kxp/im_writing_a_pseudo_academic_report_on_proof_of/