

Chia Bread Pudding: a blockchain of useful space

Mónica Jin

monicachenjin@tecnico.ulisboa.pt

Instituto Superior Técnico

Portugal

ABSTRACT

Permissionless blockchains, such as Bitcoin, are secured using Proof-of-Work (PoW) consensus algorithms. While effective, PoW has a significant energy footprint, leading to the exploration of alternative approaches like Proof-of-Space (PoSp) blockchains. PoSp blockchains offer a promising alternative, relying on disk space rather than computing power to secure the system. Current PoSp techniques generate large quantities of random data for proof generation, leading to initial computational costs and wasted disk space. This thesis introduces Chia Bread Pudding, a protocol designed to employ pre-existing data, such as user files, for proof generation within the framework of PoSp systems. Our protocol demonstrates the capacity to curtail storage wastage in PoSp blockchains by at least 50%, all while upholding system performance. This research empowers greater decentralization by broadening miner participation in the blockchain ecosystem.

KEYWORDS

Blockchain, Proof-of-Space, Proof-of-Storage, Useful Storage

1 INTRODUCTION

Blockchains are amongst the most popular technologies of recent years. A blockchain is an implementation of a distributed ledger used to record transactions across a network of nodes. It allows multiple parties to reach consensus on a single version of a record, without the need for a central authority. The use of blockchains has the potential to increase security and transparency in a wide range of industries, including finance [3], supply chain management [13], and identity verification [26].

Permissionless blockchains, also known as public blockchains, are a type of blockchain that allows anyone to participate in the system and validate transactions. These blockchains are decentralized and open to anyone, and they do not require permission to join or participate. Examples of permissionless blockchains include Bitcoin [27] and Ethereum [1]. This type of blockchain enables a fully decentralized distributed system, where no single central entity exists, but rather multiple entities work together in a protocol that enables participants to trust the protocol itself, even if they do not trust each other. This means that if any single entity fails, the system can still function as intended. However, this type of blockchain is more susceptible to Sybil attacks [12], in which a malicious actor creates multiple fake identities in order to gain disproportionate influence on the network. If the attacker is able to create enough fake identities, they may be able to gain control of a significant portion of the network and influence the outcome of the consensus process. This can lead to the attacker gaining control of the blockchain and potentially altering existing records.

Prominent blockchains like Bitcoin [18] mitigate the risk of these attacks by using Proof-of-Work (PoW), which requires participants

to expend a significant amount of computing power in order to add a new block to the chain. This makes it more difficult for an attacker to create fake identities and gain influence on the system. Unfortunately, this Sybil-proof mechanism comes at the cost of high energy consumption. The energy consumption of PoW blockchains is a source of concern, as it can contribute to greenhouse gas emissions and other environmental impacts [17, 28, 36]. In addition to the environmental impact, the high energy consumption of PoW can also make it more expensive to operate a blockchain. Participants, also known as miners, must invest in powerful computing hardware and pay for the energy required to operate it, which can drive up the cost of participating in the network.

This issue led to the research on more energy-efficient Sybil-proof mechanisms such as Proof-of-Stake (PoS) and Proof-of-Space (PoSp). In a PoS system, the miner of the next block is chosen based on their stake in the network, rather than their computational power. Recently, Ethereum [1] replaced its former PoW protocol with PoS. This decision was made in order to address the high energy consumption [24].

PoSp is another promising approach that relies on disk space, rather than computing power, to ensure the security of the system. In a PoSp system, miners must prove that they have a certain amount of storage space available by allocating a portion of it to the blockchain. For this, existing PoSp techniques rely on the generation of large amounts of random data to build proofs. The Chia Network [10] is a blockchain project based on PoSp that was launched in 2021. Chia has gained significant attention and adoption since its launch, with many people interested in its energy-efficient approach to securing the network. One concern with PoSp is that it may result in a waste of storage space. In order to participate in PoSp blockchains, miners allocate disk space by generating random data with specific cryptographic properties. This not only implies an initial substantial computational cost (which is amortized over the life of the system) but also wastes the space on disk with randomly generated data. Currently, in Chia, all miners provide around 32 billion gigabytes of storage that has no purpose other than securing the blockchain.

The goal of this work is to reduce the storage waste of PoSp blockchains. One way to reduce the storage waste of PoSp blockchains is to make the random data used for proofs useful (i.e., having another purpose for the miner beyond ensuring participation in the system). For this, we explore methodologies that uphold the guarantees of PoSp while leveraging the user's existing data, such as user files, for proof generation. This approach leads to the development of Chia Bread Pudding, a novel protocol that utilizes locally available data for proof generation in PoSp blockchains. By employing an encoding technique on miners' preexisting *useful* data, we enable them to compute the necessary proofs, thereby altering

the manner in which miners demonstrate their storage space commitment for the blockchain. Through modifications to Chia’s PoSp protocol with a PoSt protocol, our results demonstrate a significant reduction of at least 50% in Chia’s storage waste, with no substantial impact on the overall performance of the blockchain.

2 BACKGROUND & RELATED WORK

Cryptographic proofs play a fundamental role in ensuring the security and integrity of blockchains. They verify whether miners have invested in a non-counterfeitable resource, i.e., a resource that cannot be replicated or duplicated without significant cost.

2.1 Proof-of-Work

Proof-of-Work (PoW) is a cryptographic proof where the resource in question is computational processing power. It enables a network participant to prove that significant computational effort has been expended to solve a cryptographic problem. This proof is easily verifiable by other participants. PoW is a widely used cryptographic mechanism in decentralized and distributed systems. Although it has been proposed for multiple security purposes, including protection against denial-of-service attacks and spamming [14, 21, 23], it was only with the emergence of Bitcoin that its potential was recognized. Several limitations accompany PoW blockchains, including scalability issues and significant energy wastage [17, 28, 36]. To tackle these challenges, various alternatives have emerged in recent years. We list some of them in the following sections.

2.2 Proof-of-Stake

Proof-of-Stake (PoS) is a cryptographic protocol that serves to demonstrate an individual’s ownership of stakes within a blockchain network. PoS is used as a Sybil-proof mechanism that uses the assets in the system as a resource. The probability that a stakeholder is selected to add the next block is proportional to their wealth in the system. This approach has the advantage of consuming less energy because the mining process does not require intensive computation. However, PoS blockchains are at risk for *costless simulation* (also known as *nothing-at-stake*) and *long range attacks* [32].

Costless simulation attacks involve malicious miners generating a surplus of proofs beyond the protocol’s stipulated amount due to the deliberately inexpensive mining process. This can lead to manipulation of block content order, thereby enabling miners to select blocks with higher chances of chain inclusion. This also compromises the dynamic of the blockchain, since one of the purposes of this Sybil-proof mechanism is energy efficiency. Additionally, attackers may attempt to prolong their block’s validity by extending multiple chains, leading to a slowdown in consensus and potential failure to achieve convergence among nodes.

Long-range attacks¹ compromise the integrity of the ledger, allowing a group of nodes to manipulate the blockchain’s history if they possess the majority of the currency. They can create a new chain that supersedes the honest one, exploiting the advantages of their stake in the system. While also a concern in PoW blockchains, the threat is exacerbated in PoS blockchains due to the presence of costless simulation.

¹These attacks are a specialization of costless simulation attacks, but we state them separately due to their unique purpose.

A variety of PoS-based blockchain protocols have been proposed, with different properties and different ways to solve the aforementioned problems such as Snow White [11], Ouroboros [25], Algorand [19], and Ethereum [8]. However, concerns about the rich-get-richer issue in PoS-based blockchains persist, prompting researchers like Huang et al. [20] to examine the fairness of PoS incentive protocols compared to PoW models. They concluded that Ethereum is the only system capable of achieving similar fairness levels to PoW models under specific configurations.

2.3 Proof-of-Space

Proof-of-Space (PoSp) is a cryptographic proof where the non-forgable resource is storage space. Dziembowski et al. [15] first introduced this protocol as a more environmentally friendly alternative to PoW. PoSp is a protocol with two distinct phases: initialization and execution. During initialization, large amounts of random data with specific cryptographic properties are generated and stored on disk. In the execution phase, the previously stored data is used to generate a PoSp proof. PoSp has an initial computational cost that is amortized over the system’s lifetime, as the stored data can be reused for several future proofs. The initialization process of PoSp is based on a time-consuming process of generating random data, significantly more time-consuming than searching for this data stored on the disk. Therefore, in the context of blockchains, miners are incentivized to store the data rather than generate it in real-time when creating a new block for the chain. There are two approaches to PoSp.

The first approach, presented by Dziembowski et al. [15], is based on hard-to-pebble graphs and Merkle trees. Dziembowski et al. prove that an attacker uses $\Omega(N)$ space to store proofs or makes $\Omega(N)$ invocations of the hash function. In other words, an attacker who chooses not to store the appropriate space during the initialization phase will have to compute multiple hash functions when creating a new block for the blockchain. Thus, a rational participant always opts for the first option, as it is cheaper and faster. This variant of PoSp offers the best security guarantees for this type of proof but is challenging to implement. The size of the proofs is large (in the order of megabytes), and these proofs cannot be fully non-interactive when adapted for blockchains, i.e., participants cannot join at any time without prior preparation. Abusalah et al. [2] propose a second approach to PoSp. This approach is based on the inversion of hash functions. During the initialization phase, hash collisions are computed and stored on disk. In the execution phase, the proof is generated using the previously stored content. Although this approach does not have security guarantees as strong as the first approach, it has the advantage of having proofs with a small size and being non-interactive.

PoSp blockchains are also susceptible to costless simulation and long-range attacks, necessitating the implementation of supplementary mechanisms to mitigate these vulnerabilities. Chia [10] is the first blockchain to implement a consensus protocol based on the second PoSp approach. This protocol uses PoSp and Proof-of-Time (PoT). PoT is a cryptographic proof that ensures that a certain amount of time has been spent and is based on Verifiable Delay Functions (VDFs) [31, 38]. A VDF is a non-parallelizable function where the execution time is configurable and predetermined. For

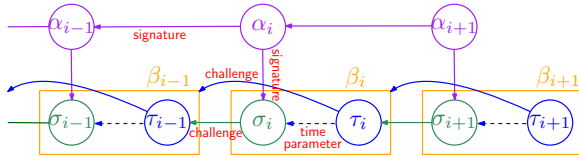


Figure 1: Illustration of the Chia blockchain, from [10]. Each block’s data α is decoupled from the proofs (PoSp σ and PoT τ).

each input, there is only one output that is quickly verifiable. In Chia, there are two types of miners, PoSp miners and PoT miners. A PoSp miner first dedicates a certain amount of space in a process called initialization. After this process, the miner can start creating blocks. To extend a chain, the miner first calculates the 256-bit challenge by obtaining the result of the hash function from the last PoT. Then the miner searches their storage for data that matches the challenge and generates a proof through this. Once the proof is generated, the miner extends their local chain with this proof and transmits it to the network. A PoT miner takes this unfinished block (i.e., a block with only a PoSp proof), validates the PoSp proof, ranks it in terms of quality, and calculates the VDF using this quality. After the VDF execution, this miner finalizes the block by extending it with a PoT proof. Chia is set to have one PoT proof every 9.375 seconds. Once each PoT is shared, PoSp miners have the opportunity to extend the blockchain, restarting the process. The blockchain is extended by alternating between PoSp and PoT proofs (illustrated in Figure 1). The runtime of the PoT is inversely proportional to the quality of the PoSp. A high-quality PoSp results in a shorter runtime for the following PoT. Chia has been widely adopted as a more accessible and environmentally friendly alternative to Bitcoin. One of the main issues with Chia is the waste of storage resources. The data generated in the initialization process can only be used to generate proofs, introducing computational and storage waste.

2.4 Proof-of-Storage

Proof-of-Storage (PoSt) [4, 7, 16, 22, 34] is similar to PoSp schemes, but instead of showing that space is allocated with random data, they prove that the allocated space is correctly storing previously provided data. PoSt is used in decentralized storage networks, as they allow participants to share and store data without relying on a single storage provider. PoSt has great potential for archiving because it allows a client to store a file on a server without trusting it and subsequently verify its integrity.

PoSts are used in blockchains as a mechanism to protect against Sybil attacks. These blockchains just like PoSp blockchains, also require mechanisms to mitigate costless simulation and long range attacks. Additionally, the work done by PoSt miners is considered useful, as the result of the computation has another utility for the network beyond ensuring the security of the blockchain. Recent examples of these blockchains include Filecoin [6] and Subspace [37].

Filecoin [6] is a decentralized storage network that turns cloud storage into an algorithmic market. This market operates with its

native cryptocurrency FIL, which miners earn by providing storage for clients. The decentralized storage network is based on a type of PoSt known as a Proof-of-Replication (PoRep) [16]. PoRep, in addition to proving that certain data has been stored by a miner, ensures that they dedicate a unique disk space and that the data is retrievable. Attackers cannot pretend to store multiple copies of the same data by deduplicating storage, ensuring that all storage providers store the replicas independently.

Subspace [37] is another PoSt blockchain where miners are incentivized to store blocks from the blockchain itself. Each miner extends the blockchain with PoRep proofs of the blocks they store.

Both Filecoin and Subspace require protocols for distributing useful data and ensuring its storage over a long period. These protocols increase the system’s complexity and, as a consequence, impact network performance. Our work seeks a simpler alternative to these protocols by removing the requirements for data distribution and retrieval.

3 CHIA BREAD PUDDING

Introducing Chia Bread Pudding, we propose a method that redefines storage optimization in PoSp blockchains. Drawing inspiration from the concept of "bread pudding protocols" [21], our approach repurposes existing storage space for practical utility, effectively reducing storage waste in PoSp blockchain protocols. Chia Bread Pudding allows miners to use their preexisting data as a non-counterfeitable resource, thereby reducing the amount of random data generated and stored for PoSp blockchains.

3.1 Challenges and Solutions

Using miners’ local files as proofs of space is not straightforward. To materialize our vision, we must address the following challenges: 1) unknown file structure and 2) low entropy.

3.1.1 Unknown File Structure. PoSp [2, 15] demonstrates the allocation of space by storing data with specific cryptographic properties. This data is generated based on cryptographic structures and incurs computational and temporal costs. User file structures are unpredictable and do not adhere to the constructions of PoSp proofs, lacking the required cryptographic properties. In addition to the absence of these properties, it is crucial that the file structure remains preserved (i.e., the file content remains unaltered and can be directly read) or recoverable (i.e., the content is modified but can be recovered and read). In other words, using miners’ files is only useful if their original state is maintained or recoverable. To deal with this challenge we replace the PoSp with a proof technique that allows us to use existing data to prove that specific storage resources were allocated for it. For this, we adapt the cryptographic techniques used in PoRep [16] to suit our requirements. PoRep guarantees the accurate replication of data across locations while confirming the allocation of equivalent storage resources. The replicated data remains retrievable, ensuring the possibility of recovering the original information from this duplicate. Consequently, PoRep functions as PoSp, validating the allocation of a designated space with useful data and assuring its recoverability. Our adaptation of PoRep to the context of allocating miners’ files for the blockchain maintains the methods for verifying the space allocated by the files, but it does not encompass integrity verification. This is attributed to each

miner’s ownership of their files, eliminating the necessity for integrity verification. Furthermore, by not verifying the content of the files we allow each miner to choose to either use preexisting data or generate random data for the blockchain. Miners are incentivized to use existing data since they do not need to allocate more resources for random data. By allowing this dynamic, we expand the domain of potential participants of our system whilst also reducing the storage waste associated with traditional PoSp blockchains.

3.1.2 Low Entropy: The data used to create PoSp proofs is generated using hash functions. If the input to these hash functions is repetitive, it will produce repetitive results. The diversity of results is what enables miners to obtain various different proofs and increase their chances of extending the blockchain. Archived files typically exhibit low entropy since various sections contain repetitive or redundant data [33]. To address this challenge, we propose encrypting the original files. When data is encrypted, it undergoes a process where it is converted into ciphertext using an encryption algorithm and a specific key. This process obscures the original information, by ensuring the underlying patterns of data remain unpredictable. This obfuscation increases the entropy of the data and ensures the privacy of miners’ data is protected. Using a cipher in Cipher Block Chaining (CBC) [5] mode increases entropy and ensures a certain level of privacy for miners’ data. Furthermore, the cipher allows for sequential encryption and parallelizable decryption. Thus, the proof initialization process remains sequential while the recovery of the original data is efficient.

3.2 Base protocol - Chia

Chia Bread Pudding redefines blockchain by focusing on the concept of useful storage allocation. Despite advances, challenges raised by Park et al.[29], such as costless simulation and long-range attacks, continue to be prominent concerns for our blockchain ecosystem. To effectively tackle these challenges head-on, we embrace Chia’s protocol [10], incorporating its proven solutions to these issues as the foundational framework of our system, thereby bolstering our defenses against potential threats. Within our blockchain, miners start by undergoing an initialization process before actively participating. Upon completion of this initialization phase, miners are equipped to engage in the mining process. They receive challenges from the last block in the blockchain, generate proofs based on data stored during initialization, and extend the chain by incorporating newly generated proofs in the new block. Subsequently, the network verifies proofs submitted by multiple miners, ultimately selecting the block with the highest-quality proof.

When adapting a PoSp blockchain like Chia, we need to ensure that the cheap process of creating a proof from the stored initial data does not lead to common risks like the ones identified by Park et al.[29] are addressed as follows:

Interactivity: Miners use their own local data to compute the initialization step. This step of our adapted PoRep does not require an initial commitment to the blockchain. Miners are able to participate in the blockchain once they finish the initialization step.

Determine the winner: Just like Chia’s original protocol, our protocol specifies a quality for each proof. The proof with the best announced quality is considered to be the extension of the chain. Similarly to Chia, the probability of a miner successfully

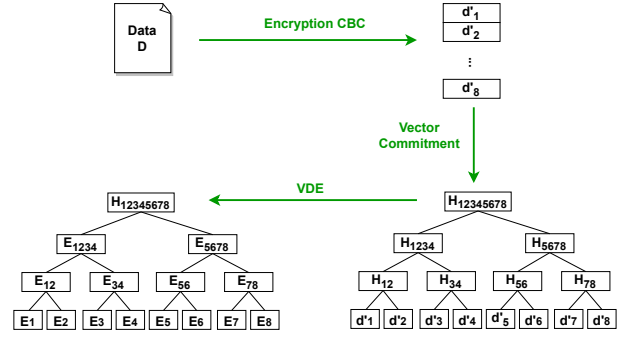


Figure 2: Initialization process

extending the chain is proportional to the space allocated, which is measured using our own proof instead of a PoSp. The winning chain is determined by the highest accumulated quality of a chain.

Costless simulation: Our proposal keeps Chia’s original methods for dealing with costless simulations. The transactions and the proofs of a block are decoupled into two separate chains, preventing attackers from tampering with each block’s content in an attempt to obtain a higher quality proof.

Long-range attacks: Chia’s original method for preventing attackers from rewriting the blockchain’s history is kept in our proposal. However, instead of alternating between PoSp and PoT proofs, our proposal replaces PoSp with our adaptation of PoRep. The PoT proofs are computed using VDFs that take a predetermined time to compute. The time spent in VDFs is defined by the quality level of each space proof. Higher-quality proofs result in smaller computing time in the VDF.

Our protocol modifies Chia’s protocol by altering the following algorithms:

- (1) Initialization (Figure 2);
- (2) Proof Generation (Figure 3)
- (3) Proof Validation (Figure 4);
- (4) Proof Quality (Equation 1).

3.3 Initialization process

Prior to a miner’s engagement in the process of block creation to extend the blockchain, a mandatory initialization phase must be executed. The initialization process, as depicted in Figure 2, is a multifaceted operation that transforms a given data into a structure possessing specific cryptographic attributes, and it is executed in three distinctive steps.

The first step involves dividing the data into blocks of a uniform and predetermined size m , followed by their encryption through the utilization of a cipher in CBC mode. As described in Section 3.1 this encryption adds a layer of obfuscation and randomization to the original data, thereby preserving data privacy and elevating data entropy.

In the second step, we employ a Vector Commitment (VC) protocol [9] to secure the encrypted blocks. A VC is a cryptographic technique that enables a party to commit to a vector of values while allowing for later individual value revelations, all while providing cryptographic proofs to verify that the revealed values correspond

to the committed ones. This commitment protocol offers efficiency in both commitment, which involves creating a commitment to the entire vector, and opening, where individual values within the vector are revealed. Importantly, both commitment and opening operations incur minimal computational overhead. In our implementation, we opt for a Merkle tree [35] as our chosen VC scheme. In this scheme, each encrypted block serves as a leaf in the Merkle tree. These leaves are grouped together by concatenating a fixed number, denoted as f , of blocks in each group. A cryptographic hash function is then applied to each group, producing a fixed-size hash value unique to the content of the data blocks within that group. These hashed values become parent nodes to those associated with the hashed group. This grouping and hashing process is applied recursively to the parent nodes until only one hash value remains, which represents the Merkle root.

In the third step, we apply a VDE [16] to encode every node within the Merkle tree. This encoding process is deliberately time-consuming and its duration is proportional to both the data size and the Verifiable Delay Encoder (VDE) time. Notably, VDEs are slow in encoding but fast in decoding. The encoding time is adjustable as it is parameterizable, while the decoding time remains relatively constant. The encoding/decoding scheme relies on VDFs [31, 38]. The encoding process is executed through a sequential and chained technique, wherein the encoding of one node is intricately tied to the encoding of other nodes, akin to the encryption process used in block-chaining cipher modes. The process starts by encoding the root node using a VDE. Then, the hash of the encoded root is used to encode each of its child nodes. All the rest of the nodes are encoded using the encodings of their parents' sibling nodes. All the nodes except for the root are encoded as the following: $VDE(node \oplus key)$. The key is the hash of the concatenation of all sibling nodes (those sharing the same parent) of the target node's parent. This iterative process continues for each subsequent child node of the target node until the leaf nodes are successfully encoded.

This initialization process can be repeated for various files. Each file is transformed into an encoded Merkle tree. Using more files during initialization increases a miner's chances of extending the blockchain due to a greater data diversity enabling more proofs.

The initialization process is intentionally configured to be time-consuming, serving as a strategic incentive for miners to retain and preserve the data generated during this crucial phase. The significant time investment required during initialization primarily results from the integration of VDE. By compelling miners to dedicate time and effort to the data generation process, our protocol reinforces a sturdy foundation for the blockchain's security, ensuring the enduring relevance of the generated data throughout the blockchain's lifecycle.

3.4 Proof generation

Proof generation is a process that enables miners to demonstrate the completion of the initialization process. It involves the creation of a proof that validates the storage of data during the initialization phase. Miners undertake this process post-initialization to actively participate in the blockchain.

A proof in our protocol is essentially an opening of the selected VC protocol, implying the disclosure of a subset of the data stored

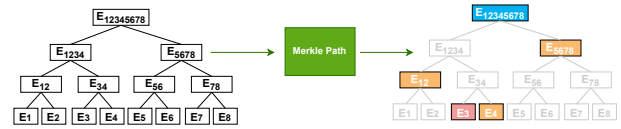


Figure 3: Proof generation

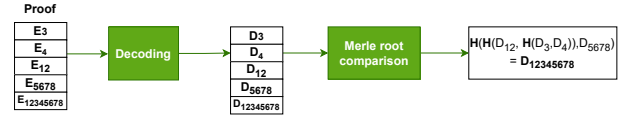


Figure 4: Proof Validation

during the initialization process. In the context of a Merkle tree, an opening corresponds to the selection of specific nodes within the tree. This selected subset of nodes includes a leaf and its corresponding Merkle path, which constitutes the minimal number of additional nodes in the tree necessary for the computation of the root hash. Each leaf of the Merkle tree possesses its corresponding Merkle path. To ensure that miners do not merely store a single valid Merkle path, we enforce a random selection of the leaf node. To achieve this, we leverage the data from the most recent block of the blockchain, facilitating the determination of the leaf node for the proof.

The proof generation process begins when a miner receives a blockchain challenge (e.g. the PoT of the latest block, represented as τ in Figure 1), which consists of a 256-bit number. The miner selects the Merkle tree with the best quality and creates a proof using the tree nodes. This quality is defined later, in Section 3.6. Proof generation involves selecting the leaf indexed by the modulus of the challenge with the total number of leaves (in Figure 3, it's leaf E3). Then, we create a Merkle tree path from the selected leaf (shown in orange in Figure 3). Finally, the selected leaf is concatenated with the Merkle tree path and the root to generate the proof (in Figure 3, $proof = E3||E4||E12||E5678||E12345678$). Once a miner concludes the proof generation, he extends the blockchain with a block containing the new proof.

3.5 Proof validation

The process of proof validation assesses whether the presented proof and its associated challenge fulfill the necessary criteria for acceptance within the blockchain. This validation encompasses the following steps:

- (1) verify cryptographic proof;
- (2) verify challenge;
- (3) verify minimum proof quality.

The first verification validates the proof using cryptography methods. This step ensures that the proof creator is committing a specific amount of data to the blockchain, confirming that the proof was indeed generated using the data stored after the initialization process. This verification, depicted in Figure 4, consists of two key steps: 1) Decoding, and 2) Merkle root computation and comparison.

In the decoding step, the receiving miner decodes each node with a Verifiable Delay Decoder (VDD), the inverse of VDE. Each node is

decoded using the hash value of its following encoded nodes. Every node except the root node is decoded as follows: $VDD(node \oplus key)$. The key value is the hash output of a subset of encoded nodes from the proof. This subset of nodes is composed of the ones used in the encoding process, therefore not every node has the same key value. Once the nodes are decoded, the process proceeds to the computation and comparison of the Merkle root. This step involves computing a new root from the decoded nodes and comparing it with the root provided in the proof. Computing a Merkle root from the decoded nodes involves an iterative hashing process, where the leaf node is hashed with a node from the Merkle path, and the result of this hash is subsequently hashed with the next node from the path. This hashing process continues until it reaches the last node of the Merkle path. The resulting output of the final hash represents the newly computed Merkle root. Finally, the computed root is compared with the root provided in the proof. The proof is deemed valid if the two roots are a match.

The second verification validates the challenge that is associated with the proof. In this regard, we adhere to the original Chia approach for validation. Specifically, a challenge is deemed valid if it falls within the category of the three most recent PoT proofs. Consequently, miners are allotted a window of 9.375-28.125 seconds to generate and share a proof for each challenge. This validation of challenges ensures that the data created during the initialization process remains stored and utilized throughout the blockchain’s lifespan. Due to the time-intensive initialization process and the limited validity period of each challenge, miners are unable to generate data on the fly and thus are compelled to retain the data generated during initialization.

Finally, and similarly to the original Chia protocol, a proof is deemed valid only if its quality value falls below the changing *difficulty* value, which evolves over the course of the blockchain’s existence. Further elaboration on the proof quality function is provided in the next subsection.

3.6 Proof Quality

Proof quality refers to a function that assigns a value to a given proof, serving the dual purpose of proof validation and ranking the proofs submitted by miners for inclusion in the blockchain. Notably, a lower output value of this function corresponds to a superior proof. Our adaptation of Chia’s initial quality function results in the following calculation for the quality value:

$$quality = \frac{difficulty * sha256(merkle_root||challenge)}{plot_constant} \quad (1)$$

Both the *difficulty* and *plot_constant* parameters are defined by the original Chia protocol and remain integral to our solution. The *difficulty* is a variable that is updated periodically to moderate the rate of proofs that are accepted by the blockchain. The *plot_constant* parameter is a constant that indicates the size of the plot, i.e. the size of the data generated in the initialization process. In Chia, it is possible to generate plots of different sizes, but in our case, we only allow for one universal size.

The quality function depends on the root of the Merkle tree given by the proof. This allows for an efficient search for proof with quality without having to look up every tree node. Furthermore, having the quality be dependent on the Merkle root encourages

miners to allocate more files to the blockchain, as each file generates a new root.

3.7 Practical Considerations

3.7.1 Original Data Retrieval. For our protocol’s storage resource allocation to be deemed useful, the ability to retrieve the original data from the stored information during the initialization phase (discussed in Section 3.3) is crucial. When a miner with files allocated to the blockchain seeks to access them, the original state of the data can be generated by decoding and decrypting the Merkle trees associated with the specific files. The decoding process mirrors the one described for proof decoding, albeit applied to all the leaf nodes of the tree rather than a subset included in a proof. Post-decoding, the miner can generate the original data by decrypting all the data of these leaves. The process of data retrieval involves generating the original data using the encoded data, without modifying the associated Merkle trees, enabling their continued use in generating proofs for the blockchain. The retrieval process does not alter the data designated for the blockchain, thus it does not impede miners’ participation in the blockchain. The process of data retrieval adds a certain level of computational overhead due to the necessity of decoding and decrypting data. Nevertheless, this overhead has a negligible effect on the performance, as the decoding and decryption procedures are designed to ensure efficiency and parallelizability.

3.7.2 Data Modification and Adaptation. Following the allocation of files to the blockchain, miners may find the need to modify and render the earlier unaltered data irrelevant. To achieve this, the miner retrieves the original data (described in the preceding section), makes modifications, removes the previous data, and reallocates the updated data to the blockchain via the initialization process. The proofs generated from the deleted data that have already been integrated into the blockchain remain valid. Once the initialization process for the new files is completed, the miner can proceed to utilize them for proof generation. Furthermore, it is worth noting that the process of modifying files and subsequently reinitializing them is not expected to occur frequently. This is because write operations are generally less frequent than read operations, which corresponds with the prevalent WORM paradigm observed in numerous storage systems. As a result, the tendency for the data allocated to the blockchain to be modified will also be infrequent, aligning with the overall usage pattern of such storage systems.

3.7.3 Optimizing Storage Allocation: Useful and Useless Data. As expounded upon in Section 3.1, Chia Bread Pudding introduces a novel approach that allows miners to allocate both useful and useless data to the blockchain. The protocol primarily focuses on the amount of storage resources designated for the blockchain, without delving into the specifics of the stored content. The initialization process described in Section 3.3 can be used either on useful data or useless data. By affording miners the discretion to choose between useful and useless data, they gain the flexibility to tailor their allocated blockchain data in accordance with their individual requirements.

3.8 Public Parameters

Our proposed protocol incorporates several adjustable public parameters that influence its performance and implementation decisions:

- (1) Block and node size, denoted as m .
- (2) Number of Merkle tree leaves, denoted as n .
- (3) Merkle tree fanout, denoted as f .
- (4) VDE execution time, denoted as t .

The tuning of these parameters has a substantial impact on the system’s efficiency and implementation choices. It is worth noting that these parameter definitions also hold sway over crucial considerations concerning Hash Function selection and VDE output sizing, with a particular emphasis on ensuring alignment with the specified value of m . In Section 4.2, we delve into a comprehensive analysis of the trade-offs associated with these parameters.

4 EVALUATION

The goal of this work is to study techniques that repurpose the data used in PoSp blockchains for some useful purpose besides securing the blockchain. To this end, we identify the following two relevant research questions:

- RQ1:** To what extent can we replace Proof-of-Space blockchain’s random data with useful data?
- RQ2:** What is the impact of repurposing the data used in Proof-of-Space blockchains?

To answer these research questions, we 1) define the evaluation metrics for all RQs; 2) collect the results on these metrics to evaluate our proposed solution.

Our evaluation consists of two parts: analytical evaluation and experimental evaluation. In the analytical evaluation, we analyze the impact of each public parameter of the protocol listed in Section 3.8 to find the best configuration. In the experimental evaluation, we run the proposed protocol with specific public parameters and collect the results.

4.1 Metrics Selection

To assess our proposed solution and address our research questions (RQs), we introduce the following set of quantitative metrics:

Useful Space Ratio: Proportion of useful space in relation to total space used for proofs. Allows evaluation of the space optimization by indicating how much space can be utilized effectively.

Proof size: The size of the proofs submitted to the blockchain directly affects the amount of data that needs to be stored and transmitted. This allows us to evaluate the performance of our system.

Initialization time: The time it takes to generate the data used for future proofs during the initialization phase. This metric measures the efficiency of our solution in terms of time.

Proof Entropy: Diversity of the generated proofs. This allows us to assess if our solution is able to produce a high variety of proofs.

Throughput: The number of valid transactions per second that our system is capable of handling. It allows us to measure the efficiency and capability of our system.

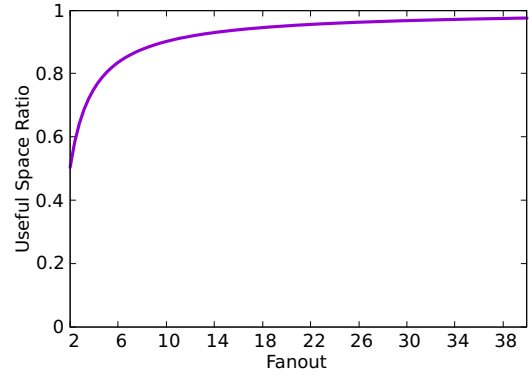


Figure 5: Useful space ratio as a function of the Merkle tree fanout for $n = \{64, 128, 256, 512, 1024\}$.

4.2 Analytical Evaluation

In the analytical evaluation section, we aim to gain an understanding of how individual public parameters affect the overall protocol. Through analysis and mathematical modeling, we deduce expressions that capture how these parameters influence the protocol’s efficiency and effectiveness in repurposing storage. This evaluation studies the relationship between individual public parameters within the protocol and its performance metrics, including useful space ratio, proof size, and initialization time. These mathematical insights provide guidance for parameter tuning, contributing to the refinement and enhancement of our blockchain solution.

4.2.1 Useful Space Ratio. The ratio of useful space is contingent upon the n number of leaves in the Merkle tree, and the tree’s fanout f . The ratio, defined as $\text{Ratio} = \frac{\text{Useful Space}}{\text{Total Space}}$, is expressed by the expression:

$$R(n, f) = \frac{n(f-1)}{fn-1}$$

Here, the Total Space refers to the entirety of space occupied by the Merkle tree, encompassing all its nodes. On the other hand, the Useful Space refers specifically to the space occupied by the leaves of the Merkle tree.

Examining this expression and its graphical representation in Figure 5, we observe that higher fanout values lead to an increased useful space ratio. This phenomenon is due to the reduction in the number of intermediate nodes in the Merkle trees as the fanout increases. Since these intermediate nodes are not considered part of the useful space, their reduction effectively decreases the total space, ultimately enhancing the overall ratio.

4.2.2 Proof size. The size of the proof is determined by the number of leaves in the Merkle tree n , the tree’s fanout f , and the block or node size m . The proof size is given by the expression

$$S(m, n, f) = m((\log_f n)(f-1) + 2)$$

Analyzing the proof size expression and the accompanying graphical representation in Figure 6, it becomes apparent that the proof size escalates with both the fanout (horizontal axis in the figure) and the number of tree leaves (as indicated by the various functions in the figure).

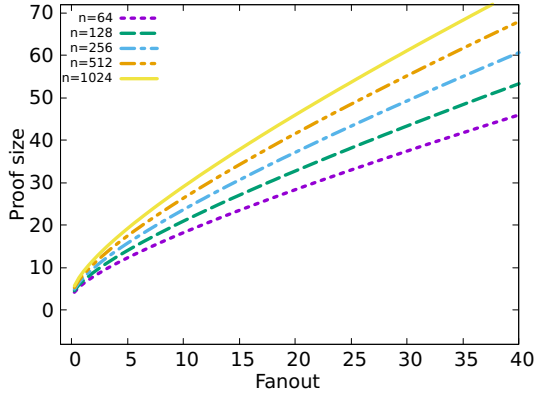


Figure 6: Proof size (multiple of m) as a function of the Merkle tree fanout for $n = \{64, 128, 256, 512, 1024\}$.

4.2.3 Initialization time. The initialization time is contingent on the execution time of the VC $v(m, n, f)$, the execution time of the chained encoding $e(m, n, f)$, the execution time of the VDE t , and the execution time of the CBC $c(m, n)$. The initialization time is given by the expression:

$$T(n, f, t) = c(m, n) + v(m, n, f) + t \times e(m, n, f)$$

The expression for the initialization time emphasizes its augmentation with the increase in VDE execution time and the data’s size, i.e., m and n .

4.2.4 Parameter tuning. Parameter tuning represents a critical aspect in shaping both the performance and security of the system.

For optimal system performance, we aim to minimize proof size and increase the useful space ratio. The proof size influences the system’s complexity, impacting both block transmission within the chain and blockchain storage by miners. The useful space ratio directly reflects the utility of the data being stored by miners. If this ratio is too small, the system approaches current PoSp blockchains where stored data serves no purpose beyond system security. Therefore, the public parameters should be tuned for a high ratio and low proof size to provide more utility to the stored data and establish a more sustainable dynamic for blockchains. For this, f should be high enough to provide more useful space but not high enough that it generates a proof size that allocates a great portion of the block.

Parameter tuning is a critical aspect influencing the system’s overall security. The temporal and spatial resources invested in the initialization phase hold a direct bearing on the protocol’s security robustness. When the time and space allocation during the initialization step are not cost-prohibitive, miners lack the necessary incentives to adhere to the original protocol, where data is processed and stored on disk for subsequent reuse. Additionally, the scale of data utilized for initialization must be sufficiently substantial to deter any attempts at parallelization. The data employed for each initialization process remains independent of the data utilized in other initiation processes. Thus, if the initialization data is not on a significant scale, miners can potentially parallelize multiple initiation processes, resulting in reduced overall time spent on each individual process.

Table 1: Execution time in minutes for the initialization process for 6, 12, 25, 51, 105 GB.

| Protocol / Generated data(GB) | 6 | 12 | 25 | 51 | 105 |
|-------------------------------|------|-------|-------|-------|--------|
| Chia | 68.9 | 130.3 | 252.5 | 514.3 | 1352.2 |
| Chia Bread Pudding | 72.8 | 142.5 | 273.1 | 528.6 | 1171.8 |

We discuss the concrete values used in our experimental evaluation in the next section.

4.3 Experimental Evaluation

In the experimental evaluation, we aim to evaluate our blockchain solution through empirical observations. In this section, we tune our protocol’s public parameters in order to achieve similar performance results as Chia’s original protocol. We present the results obtained for each evaluation metric from running both our developed protocol and the original Chia protocol on the same setup.

To experimentally evaluate the developed protocol, we ran the implementation of our protocol alongside the original Chia initialization protocol², which served as our baseline reference.

All the experiments were run in a machine with an Intel(R) Xeon(R) Silver 4116 CPU 2.10GHz, with 64GB of RAM.

User data was generated using the DEDISbench generator [30] in "Personal Files" mode, which follows the distribution of real user storage systems.

The public parameters used in the evaluation were as follows: $m = 256$ bits; $n = 2^{18} = 262144$; $f = 2$; $t = 0.5s$.

4.3.1 Useful Space Ratio. The useful space ratio in our experimental setup can be computed using the formula outlined in Section 4.2. For the given values of $m = 256$ and $f = 2$, the resulting useful space ratio stands at 50%. This marks a substantial enhancement when compared to Chia’s original protocol, which achieves a 0% useful space ratio. Moreover, this improvement translates into a 50% reduction in the free space required on the miner’s side, effectively decreasing storage demands.

4.3.2 Proof size. The proof size for our protocol with the defined parameters is 640 bytes. Chia’s proof size is 256 bytes. Our proof is 2.5 times larger than Chia’s.

Having a larger proof leads to more data being transmitted and stored in the network since a larger proof leads to a larger block in our blockchain. But considering that the maximum block size in Chia is 400KB, our protocol’s proof size remains practical, as it is only 0.16% of a block.

4.3.3 Initialization time. Table 1 lists the initialization execution time in minutes for data of various sizes (6, 12, 25, 51, and 105 GB). These sizes were selected through Chia’s space parameter k , which controls the size of each proof and the allocated disk space.

The execution times for each protocol are similar for the defined public parameters. Our solution has slightly higher execution times than Chia. However since the initialization process is only run once for each initial data, the difference in initialization times does not impact the security or the blockchain’s performance

²<https://github.com/Chia-Network/chiapos>

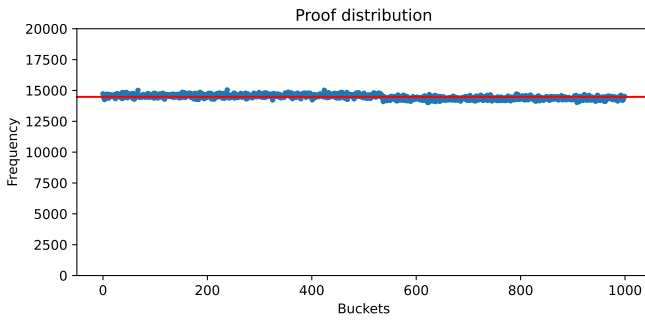


Figure 7: Proof distribution of 105GB of data generated during the initialization process(in blue) and the uniform distribution(in red).

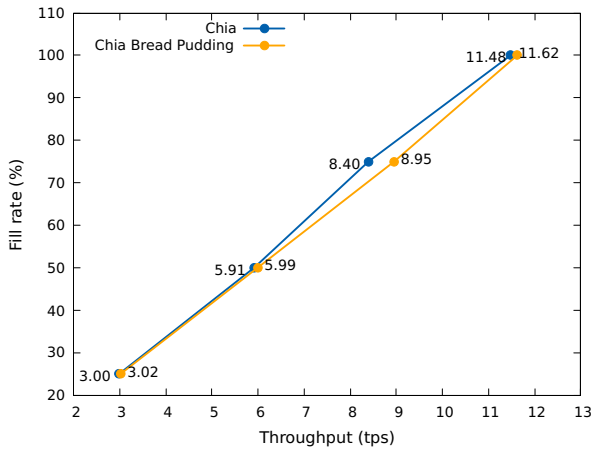


Figure 8: Throughput for different fill rates.

4.3.4 Proof Entropy. To analyze the entropy of the proofs, we compared the distribution obtained from the proofs generated from 105GB with a uniform distribution.

To obtain the distribution of the proofs, we considered each proof as a hexadecimal number categorized from 0 to 1000 and recorded the frequency of each category (Figure 7). Applying the Pearson chi-squared test to the proof distribution, we obtained a p-value of 6.75×10^{-57} , indicating that the proofs generated by our protocol are uniformly distributed. This implies that the generated proofs have high entropy.

Having proofs of high entropy is important for the security of the protocol because it means that no proof is more likely to occur than others. Furthermore, this incentivizes miners to allocate more data for the initialization process in order to obtain more diverse proofs.

4.3.5 Throughput. To measure the throughput of our proposed blockchain compared with Chia’s we ran private a simulation with 25 nodes operating as PoSp miners, alongside 1 PoT miner, a specialized node responsible for supplying PoT proofs. In the context of our simulation, introducing multiple PoT miners did not yield

any discernible alteration in the blockchain’s performance. This phenomenon is attributable to the fact that, within our system and in Chia’s, only the initial PoT received by each miner is utilized. Consequently, any additional proofs furnished by other Timelords are disregarded, having no impact on the overall performance.

Conducting assessments with a substantial miner count necessitates an extensive storage capacity, an impractical feat in our experimental context. Consequently, in our simulation, each miner possesses an individual proof exclusively designated for the generation of all blocks.

In our workload scenario, synthetic transactions were employed, with each node generating a specific quantity of these transactions, all slated for inclusion in each block. We conducted numerous simulations, each lasting an hour, with variations in the transaction volume.

Notably, Chia’s protocol prescribes a maximum limit of 1010 transactions per block. For our evaluation, we introduce the concept of “fill rate,” representing the proportion of transactions within a block relative to this predetermined upper limit (i.e., 1010 transactions).

In Figure 8, we can observe the transactions per second achieved at various fill rates: 25%, 50%, 75%, and 100%. Interestingly, the throughputs attained in both Chia and our protocol exhibit remarkable similarity. This outcome aligns with our expectations, given that the Timelord predominantly governs the throughput of both protocols. The PoT miner finalizes each block generated by PoSp miners with a PoT proof, a process that consumes a fixed amount of time for generation. Furthermore, the periodic release of challenges to miners by the PoT miner ensures that the overall system’s throughput remains relatively constant. Consequently, the observed throughput figures closely mirror each other in both Chia and our protocol.

4.4 Discussion

Answer to Q1: To what extent can we replace Proof-of-Space blockchain’s random data with useful data? Our findings affirm that replacing random data within PoS blockchains with useful data is indeed feasible. However, the extent of this substitution is circumscribed by the metadata generated during the initialization of the useful data. These metadata serve dual functions: facilitating the recovery of the original data and verifying the allocated space. In our protocol, the magnitude of the metadata is intrinsically linked to the designated fanout value. At a minimum, our framework permits a 50% ratio of useful data. Notably, this ratio can be amplified by opting for higher fanout values, as elucidated in Figure 5. It is imperative to recognize that augmenting the fanout also induces an enlargement in proof size. Therefore, the fine-tuning of this parameter necessitates judicious consideration to maintain these metrics within practical bounds.

Answer to Q2: What is the impact of repurposing the data used in Proof-of-Space blockchains? The utilization of useful data for the creation of proofs enables miners with limited storage capacity to actively participate in the blockchain, thereby expanding the pool of miners and fostering greater decentralization within the system. Moreover, by repurposing data in PoSp blockchains, we mitigate the inherent storage resource waste associated with

these systems. Nonetheless, it is worth noting that the size of PoRep proofs often surpasses that of practical PoSp proofs. This increase in proof size can lead to longer block transmission times within the system and greater blockchain storage requirements. However, it is important to emphasize that the public parameters of our protocol can be fine-tuned to mitigate impractical block transmission times and storage overhead. In practical terms, the increase in proof size remains marginal when compared with the Chia block size.

5 CONCLUSIONS

In PoSp blockchains, miners fill their storage space by generating cryptographic data randomly, solely for the purpose of ensuring blockchain security. These techniques result in storage wastage, as the stored data serves no purpose for the miner other than generating cryptographic proofs.

This work proposes an approach to reduce the storage waste in PoSp blockchains by utilizing local data (such as user files) to generate proofs. The solution is based on the Chia protocol and adapts it by replacing random data in the proofs with useful data.

Our results show that at least 50% of the random data in PoSp proofs can be replaced with useful data. Our proposed solution does not compromise the performance of the system but fosters greater decentralization within it by expanding the pool of miners who are able to participate in it.

REFERENCES

- [1] 2022. The merge. <https://ethereum.org/en/upgrades/merge/>
- [2] Hamza Abusalah, Joël Alwen, Bram Cohen, Danylo Khilko, Krzysztof Pietrzak, and Leonid Reyzin. 2017. Beyond Hellman’s time-memory trade-offs with applications to proofs of space. In *Advances in Cryptology—ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II 23*. Springer, 357–379.
- [3] Omar Ali, Mustafa Ally, Yogesh Dwivedi, et al. 2020. The state of play of blockchain technology in the financial services sector: A systematic literature review. *International Journal of Information Management* 54 (2020), 102199.
- [4] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary Peterson, and Dawn Song. 2007. Provable data possession at untrusted stores. In *Proceedings of the 14th ACM conference on Computer and communications security*. 598–609.
- [5] Mihir Bellare, Joe Kilian, and Phillip Rogaway. 1994. The security of cipher block chaining. In *Annual International Cryptology Conference*. Springer, 341–358.
- [6] Juan Benet and Nicola Greco. 2018. Filecoin: A decentralized storage network. *Protoc. Labs* (2018), 1–36.
- [7] Kevin D Bowers, Ari Juels, and Alina Oprea. 2009. Proofs of retrievability: Theory and implementation. In *Proceedings of the 2009 ACM workshop on Cloud computing security*. 43–54.
- [8] Vitalik Buterin. 2013. Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform. <https://github.com/ethereum/wiki/wiki/White-Paper>
- [9] Dario Catalano and Dario Fiore. 2013. Vector Commitments and Their Applications. In *Public-Key Cryptography – PKC 2013*, Kaoru Kurosawa and Goichiro Hanaoka (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 55–72.
- [10] Bram Cohen and Krzysztof Pietrzak. 2019. The chia network blockchain. *vol 1* (2019), 1–44.
- [11] Phil Daian, Rafael Pass, and Elaine Shi. 2019. Snow white: Robustly reconfigurable consensus and applications to provably secure proof of stake. In *International Conference on Financial Cryptography and Data Security*. Springer, 23–41.
- [12] John R Douceur. 2002. The sybil attack. In *Peer-to-Peer Systems: First International Workshop, IPTPS 2002 Cambridge, MA, USA, March 7–8, 2002 Revised Papers 1*. Springer, 251–260.
- [13] Pankaj Dutta, Tsan-Ming Choi, Surabhi Somani, and Richa Butala. 2020. Blockchain technology in supply chain operations: Applications, challenges and research opportunities. *Transportation research part e: Logistics and transportation review* 142 (2020), 102067.
- [14] Cynthia Dwork and Moni Naor. 1992. Pricing via processing or combatting junk mail. In *Annual international cryptology conference*. Springer, 139–147.
- [15] Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. 2015. Proofs of space. In *Annual Cryptology Conference*. Springer, 585–605.
- [16] Ben Fisch. 2018. Poreps: Proofs of space on useful data. *Cryptology ePrint Archive* (2018).
- [17] Ulrich Gellersdörfer, Lena Klaaßen, and Christian Stoll. 2020. Energy consumption of cryptocurrencies beyond bitcoin. *Joule* 4, 9 (2020), 1843–1846.
- [18] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. 2015. The bitcoin backbone protocol: Analysis and applications. In *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 281–310.
- [19] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th symposium on operating systems principles*. 51–68.
- [20] Yuming Huang, Jing Tang, Qianhao Cong, Andrew Lim, and Jianliang Xu. 2021. Do the rich get richer? Fairness analysis for blockchain incentives. In *Proceedings of the 2021 International Conference on Management of Data*. 790–803.
- [21] Markus Jakobsson and Ari Juels. 1999. Proofs of work and bread pudding protocols. In *Secure information networks*. Springer, 258–272.
- [22] Ari Juels and Burton S Kaliski Jr. 2007. PORs: Proofs of retrievability for large files. In *Proceedings of the 14th ACM conference on Computer and communications security*. 584–597.
- [23] Aris Jules and John Brainard. 1999. Client-puzzles: a cryptographic defense against connection depletion. In *Proc. Network and Distributed System Symp.(NDSS’99)*. 151–165.
- [24] Elie Kapengut and Bruce Mizrach. 2023. An Event Study of the Ethereum Transition to Proof-of-Stake. *Commodities* 2, 2 (2023), 96–110. <https://doi.org/10.3390/commodities2020006>
- [25] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynkov. 2017. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual international cryptology conference*. Springer, 357–388.
- [26] Yang Liu, Debiao He, Mohammad S Obaidat, Neeraj Kumar, Muhammad Khuram Khan, and Kim-Kwang Raymond Choo. 2020. Blockchain-based identity management systems: A review. *Journal of network and computer applications* 166 (2020), 102731.
- [27] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review* (2008), 21260.
- [28] Karl J O’Dwyer and David Malone. 2014. Bitcoin mining and its energy footprint. (2014).
- [29] Sunoo Park, Albert Kwon, Georg Fuchsbauer, Peter Gaži, Joël Alwen, and Krzysztof Pietrzak. 2015. Spacemint: A cryptocurrency based on proofs of space. *Cryptology ePrint Archive* (2015).
- [30] Joao Paulo, Pedro Reis, Jose Pereira, and Antonio Sousa. 2012. Dedisbench: A benchmark for deduplicated storage systems. In *OTM Confederated International Conferences “On the Move to Meaningful Internet Systems”*. Springer, 584–601.
- [31] Krzysztof Pietrzak. 2018. Simple verifiable delay functions. In *10th innovations in theoretical computer science conference (itcs 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- [32] Andrew Poelstra et al. 2014. Distributed consensus from proof of stake is impossible. *Self-published Paper* (2014).
- [33] Calicrates Policroniades and Ian Pratt. 2004. Alternatives for Detecting Redundancy in Storage Systems Data. In *2004 USENIX Annual Technical Conference (USENIX ATC 04)*. USENIX Association, Boston, MA. <https://www.usenix.org/conference/2004-usenix-annual-technical-conference/alternatives-detecting-redundancy-storage-systems>
- [34] Hovav Shacham and Brent Waters. 2008. Compact proofs of retrievability. In *International conference on the theory and application of cryptology and information security*. Springer, 90–107.
- [35] Michael Szydlo. 2004. Merkle tree traversal in log space and time. In *Advances in Cryptology—EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23*. Springer, 541–554.
- [36] Sveinn Valfells and Jón Helgi Egilsson. 2016. Minting Money With Megawatts [Point of View]. *Proc. IEEE* 104, 9 (2016), 1674–1678. <https://doi.org/10.1109/JPROC.2016.2594558>
- [37] Jeremiah Wagstaff. [n. d.]. <https://subspace.network/technology>
- [38] Benjamin Wesolowski. 2019. Efficient verifiable delay functions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 379–407.