

# Stashware: A Secure and Permanent Decentralized Storage Platform for Borderless Economy

June 26, 2020

Version 0.0.1

## Abstract

We present a design for a decentralized storage network called Stashware which resists the attempts of powerful adversaries to find or destroy any stored data. We use the powerful cryptographic primitives and consensus mechanisms. We utilize both *Proof of Capacity* (PoC) and *Proof of Access* (PoA) as consensus mechanisms. Miners mine coins from the block and get block reward. Also they get rewards from storage fee and transaction fees. But after mining a block, we require them to stake based on the amount of blocks they mine. We enumerate distinct notions of security for each party in the system, and suggest a way to classify anonymous systems based on the kinds of security provided. Stashware ensures the availability of each document for a permanent or publisher-specified lifetime. Stashware's incentives-based approach is tailored such that it is simpler and more versatile, performing flexibly in both well-functioning and poorly-functioning environments. A reputation system also provides server accountability by limiting the damage caused from misbehaving servers. We also introduce and improve the flaws of existing schemes such as Filecoin, Siacoin, Storj, Lightstreams and Arweave. We identify attacks and defences against decentralized storage services, and close with a list of problems which are currently unsolved.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Decentralization of Storage</b>	<b>3</b>
<b>3</b>	<b>Architecture of Stashware</b>	<b>4</b>
3.1	Transaction Structure, IPFS, and File Management . . . . .	6
<b>4</b>	<b>Existing Consensus Mechanisms</b>	<b>7</b>
4.1	Proof of Work (PoW) . . . . .	7
4.2	Proof of Capacity (PoC) . . . . .	8
4.2.1	Advantages of PoC . . . . .	9
4.2.2	Disadvantages of PoC . . . . .	9
4.3	Proof of Stake (PoS) . . . . .	9
4.4	Proof of Access (PoA) . . . . .	10
<b>5</b>	<b>Consensus Algorithms of Stashware</b>	<b>10</b>
5.1	iPoC in Stashware . . . . .	11
5.2	Proof of Access (PoA) . . . . .	12
<b>6</b>	<b>Security and Privacy</b>	<b>12</b>
6.1	Security and Privacy Services of Stashware . . . . .	13
6.2	Secure Deduplication . . . . .	14
6.3	Public Auditing . . . . .	15
6.4	A Privacy-Preserving Public Auditing . . . . .	15
<b>7</b>	<b>Related Work</b>	<b>16</b>
7.1	Siacoin . . . . .	16
7.2	Storj . . . . .	17
7.3	Filecoin . . . . .	17
7.4	Lightstreams . . . . .	18
7.5	Arweave . . . . .	18
7.5.1	Weaknesses of Arweave . . . . .	19
<b>8</b>	<b>Development</b>	<b>19</b>
<b>9</b>	<b>Future Work</b>	<b>20</b>
	<b>References</b>	<b>20</b>

# 1 Introduction

In today's world, with development of computerized data encompassing us, people can easily convince that since data is readily accessible online today, it can't be manipulated or disappear. Most of the information right now accessible through the Internet is quite centralized and is put away with a modest bunch of innovation companies that have the involvement and capital to construct enormous information centers capable of handling this tremendous sum of data. Unfortunately, this is a quite optimistic view and may not be the case in the near future. This is because of the fact that the web may be a fiercely effective framework of disseminated data dispersal, it right now needs a complementary framework of decentralized, lasting information capacity.

Storage is defined as the retention of retrievable data on a computer or other electronic system. We use storage on a daily basis from the mobile phone and computers we use and it is easily understood from the files we put onto a USB stick. From the days of having to put files on a floppy disk to being able to place files in the 'cloud', storage has come a long way. Nowadays, the total estimated storage capacity of the Internet is more than  $10^{24}$  bytes, which equals to 1 million exabytes or 1000 zettabytes. A byte is a data unit comprising 8 bits, and is equal to a single character in one of the words people are reading now. An exabyte stores 1 billion billion characters. A couple of the challenges confronted by data centers are: information breaches, periods of inaccessibility on a amazing scale, capacity costs, and growing and updating rapidly sufficient to meet client request for speedier information and bigger groups.

Decentralization is understood as the transfer of authority from a central entity to a more localised and 'liberal' system. The concept itself has been around for awhile and an earlier concept could be paralleled to the introduction of the Internet where the spread of information was democratised. The term is now being coined against Blockchain technologies and applications such as Bitcoin and Ethereum which decentralise financial transactions and computing power.

Nearly all of the pages making up the internet nowadays are resident inside centralised information stores, each ordinarily controlled by one association or indeed one person. This implies that when getting to data online, we are entirely dependent on these centralised associations and people proceeding to permit us to do so.

In order for an data store to be really lasting, it must be both versatile and decentralized. Blockchain innovation has much self-evident guarantee within the zone of strong, decentralized data conservation, as a key include of the innovation is that all information interior the blockchain is immutable, and cannot be changed once it is put away. On the other hand, traditionally, such innovation seriously needs versatility which clearly limits its utility for putting away significant amounts of information.

Stashware targets to make the Internet available as an enormous distributed and disseminated network comprising of decentralized devices that are not controlled by a single entity or authority.

## 2 Decentralization of Storage

A blockchain-based decentralized storage system has gained increasing attention of the public because of its minimum cost for storage, maintenance, bandwidth, and services without dependence on a specific vendor. The evolution of Web 3.0 also requires a decentralized storage system of the web content rather than a centralized one. The nodes in the decentralized storage network can contribute their own disk space to store data for users/clients. Data and meta-data are encrypted and the ciphertexts are distributed to anonymous nodes in the network. Therefore, security and privacy can also be improved by the hiding of location.

In general, only metadata is stored on-chain which is protected by the owner's private key. This is because of the large volume of the data. The entire of data is stored in a feasible location among the nodes of the blockchain network which can be chosen by the data-owner (e.g., following the regulatory compliance). The location information is put on-chain as a part of the metadata which is also encrypted by the data owner. As soon as the owner wishes to retrieve his data, he can use the corresponding metadata from the blockchain, decrypt with his private key to reveal the data location, and finally can download the data from those nodes in that specific location.

### 3 Architecture of Stashware

In Stashware, the data is distributed into different chunks and is stored across different nodes of a blockchain based network. The ultimate goal of Stashware is to provide internet service which is decentralized, censorship-resistant, and permanent liveness of the data. In a decentralized system, even if some of the nodes goes down, the rest of the network will be alive to keep the data safe.

The Stashweb is a layer built on top of the Stashware's universal permanent hard drive. When a Stashware node receives a new transaction sent from an edge user or Stashweb application, the node validates the transaction amount and that the previous transaction reference matches that which is found in the Wallet List.

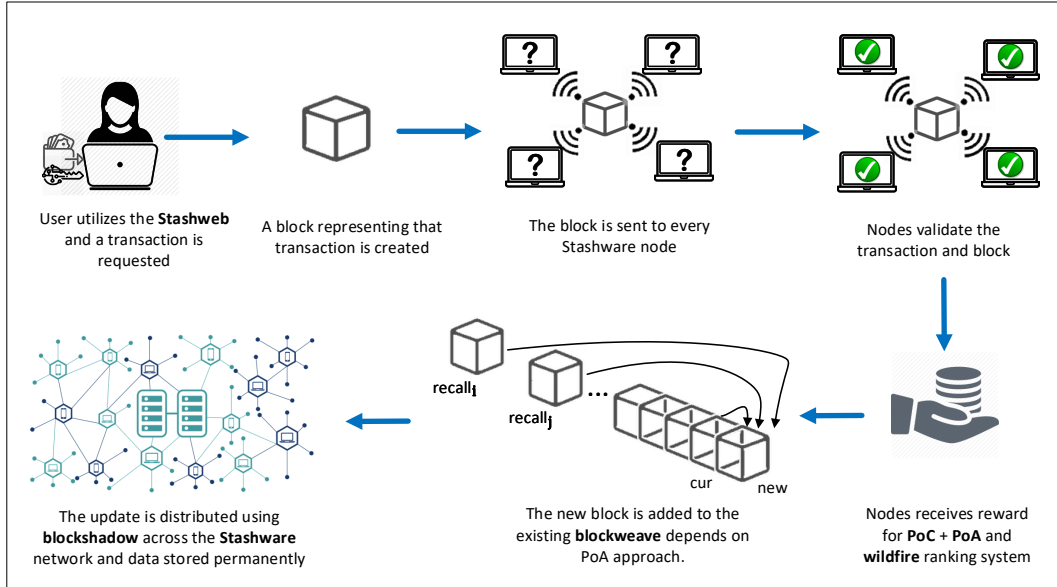


Figure 1: The flow of Stashware network.

For validation of a block PoC and PoA based consensus mechanism is used (see Section 5). In case a node successfully validates the transaction, then propagates the transaction to other nodes as soon as possible to be rewarded by the incentive mechanisms. Stashware encourages the nodes in a ranking system. As a data storage system, Stashware not only requires to be able to store large amounts of data, but also needs to provide access to the information in the most convenient way possible. Stashware creates a ranking system locally on each node, with

each peer ranked by its rank, and the overall network blacklists its peers for poor performance. Miners are encouraged to maintain a high reputation to get more reward.

It is in a node's interest to propagate a transaction straightforwardly upon receipt rather than just mining it into a block. This is because the individual transaction needs to be accepted as valid by a majority of other nodes in the network before a block containing that transaction can be accepted. The steps for validating a transaction before entry into the transaction pools are as follows:

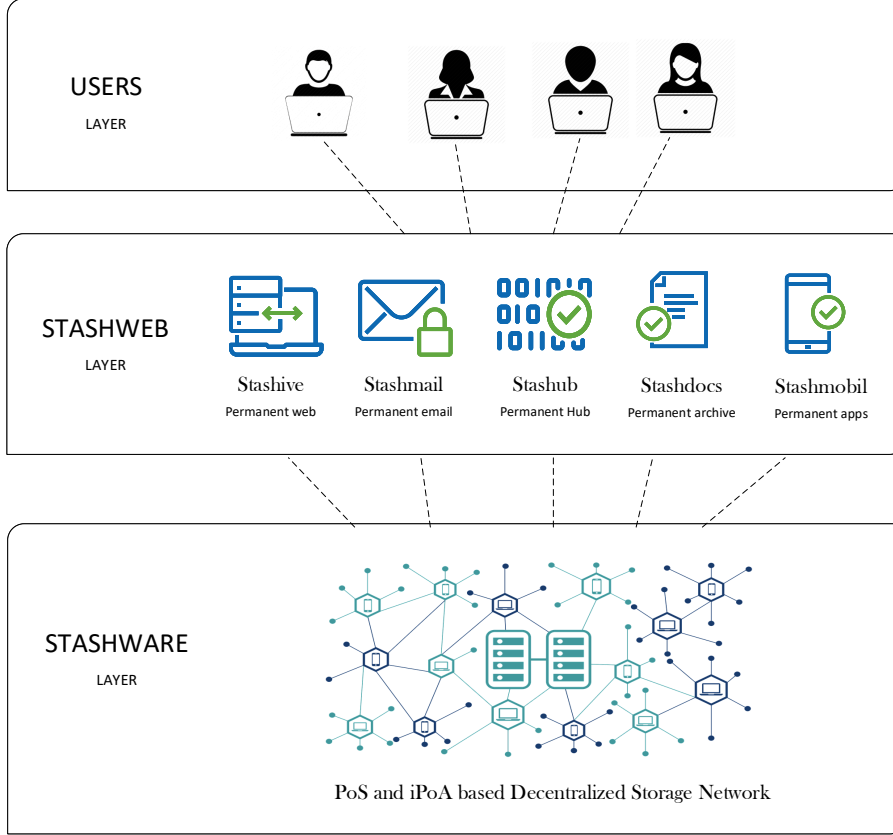


Figure 2: The architecture of Stashware and Stashweb for decentralized storage network.

1. Transactions that have already been processed are dropped.
2. Transactions that are not well-formed are ignored by the receiver node.
3. The wallet associated with the transaction must contain a sufficient stash balance in order to process claimed transaction and any additional pending transactions from the same wallet.
4.  $TX_{sender}$  and  $TX_{receiver}$  should not refer to the same wallet.
5. The transaction fee must be higher than a minimum variable fee.

6.  $TX_{anchor}$ :  $TX_{sender}$ 's last processed transaction ID or the independent hash of one of the last 20 blocks. Empty for the first transaction.
7. The  $TX_{anchor}$  should be present in the wallet list of current position as  $TX_{sender}$ 's last processed transaction ID, the independent hash of one of the last 20 blocks, or be empty for the first transaction.
8. In case of an unfriendly accessibility, the nodes will get a lower score.
9. Stashware will support both IPv4 and IPv6.
10. Inter-node protocol based on RESTful API.

### 3.1 Transaction Structure, IPFS, and File Management

Stashware adopts account based model for its transaction structure and it also support smart contract for decentralized application. For simplicity, a transaction would contain following fields depicted in Table 1.

Field	Description
Version	Version number of protocol
From	Contract deployer
To	Outgoing funds (optional)
Gas Limit	Larger enough for contract deployment (optional)
Gas Price	Determined by transaction initiator (optional)
Meta Data	Metadata or some arbitrary data
Signatures	Signature from input funds

Table 1: The content of a Stashware transaction

Interplanetary File System (IPFS) is a distributed file system that seeks to decentralize the web and to make it faster, more reliable and efficient. IPFS stores each file as an object that contains data and links. The object storing size is limited to 128 Kb. Files that are larger than 128 Kb are divided into small pieces (chunks) that are equal to or less than 128Kb. Every chunk is identified by a hash and linked to other chunks to derive Merkle Tree hash (See Fig 6).

A *file contract* is an agreement between a IPFS provider and the end users with the use of smart contract on Stashware (See Fig 3). The digital contents of files are stored on the IPFS and the IPFS hashes (merkle hash tree) are stored into the blockchain smart contracts to provide traceability and authenticity. To construct a file's Merkle root hash, the file is split into fragments of constant size and hashed into a Merkle tree. The root hash, along with the total size of the file, is used to verify the existence of storage. More specifically, the hash generated on storing the documents to IPFS, can be stored in the smart contracts effectively and documents can be accessed using the hash. If there is any change in the content of the digital document, the hash changes, to show that the original content was modified and altered.

In smart contracts, it can also be specified a duration, challenge frequency, and payout parameters, including the reward for a valid proof, the reward for an invalid or missing proof, and the maximum number of proofs that can be missed. The challenge frequency specifies how often a storage proof must be submitted, and creates discrete challenge windows during which a host must submit storage proofs (one proof per window). Contracts define a maximum

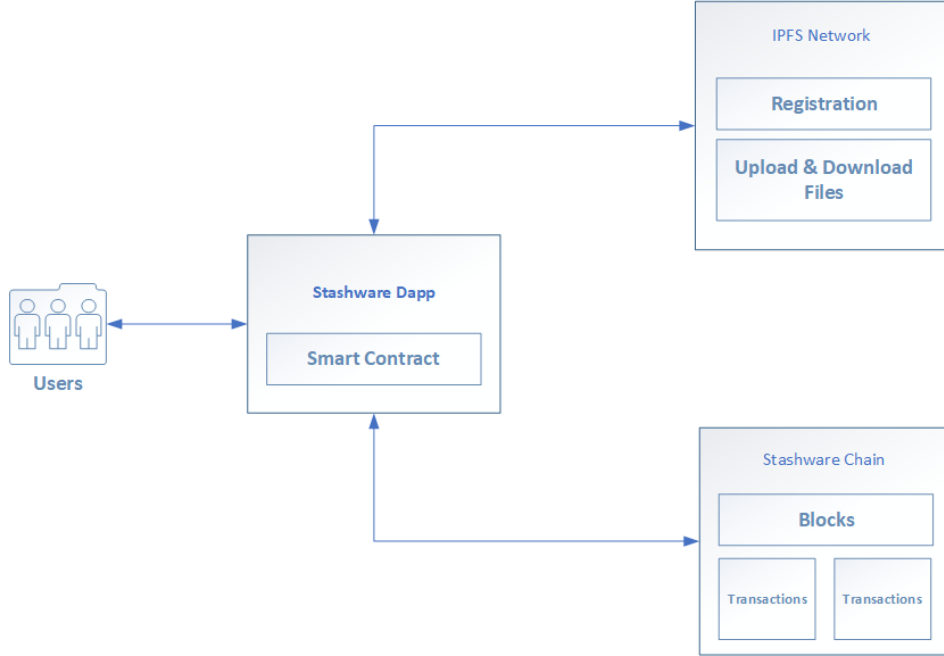


Figure 3: File Management on IPFS based on Stashware Smartcontract.

number of proofs that can be missed; if this number is exceeded, the contract becomes invalid. If the contract is still valid at the end of the contract duration, it successfully terminates and any remaining coins are sent to the valid proof address. Conversely, if the contract funds are exhausted before the duration elapses, or if the maximum number of missed proofs is exceeded, the contract unsuccessfully terminates and any remaining coins are sent to the missed proof address.

## 4 Existing Consensus Mechanisms

### 4.1 Proof of Work (PoW)

The first concept of *proof of work* (PoW) is first appeared in [10] into the usage of a cryptographic computation as a spam detection and distributed denial-of-service attack (DDoS) protection mechanism. Later this idea is utilized for a basic data channel to carry a *strong economic signal*, allowing a receiver to make a physical assertion without having to rely upon *trust* [20]. In 2008, Nakamoto’s Bitcoin white paper [13] used this idea which allows the trustless and distributed consensus mechanism. A trustless and distributed consensus system means that when someone wants to send/receive money to/from one that not need to trust in third-party services. The security model proposed by the PoW was augmented with digital signatures and a ledger in order to ensure that the historical record could not be corrupted and that malicious actors could not spoof payment. With the idea of PoW, Bitcoin became the first widely adopted global decentralised transaction ledger.

PoW can be seen as a “mathematical puzzle” that has to be quite hard to solve but easy to check for the network. This idea is also known as a CPU cost function, client puzzle,

computational puzzle or CPU pricing function.

Every node in the network races to be the first to find a solution for the puzzle. This puzzle is the candidate block, a problem that cannot be solved in other ways than through brute force so that essentially requires a huge number of attempts. When a node (miner) who finds the correct solution announces it to the whole network at the same time to get a reward for that cryptocurrency prize declared by the rules of the protocol. This puzzle in most case is an operation of inverse hashing that determines a number (nonce), so the cryptographic hash algorithm of block data results in less than a given threshold. However this mechanism PoW is energy intensive. It's costly and requires plenty of computing power. It is vulnerable to the the 51% attack. Namely 51% attacking nodes/miners could capture the network and gain dominance, thereby causing a failure in the decentralized mechanism.

## 4.2 Proof of Capacity (PoC)

Proof of Capacity (PoC) allows the mining devices in the network to use their available hard drive space to decide the mining rights, instead of using the mining device's computing power (as in the proof of work algorithm) or the miner's stake in the cryptocurrencies (as in the proof of stake algorithm). BurstCoin [17] practically used PoC in their consensus instead of PoW. We adopts BurstCoin's approach on PoC for mining a block in the chain.

PoC involves two parts. There is the plotting of the hard drive and the actual mining of the blocks. Depending on the size of your hard drive, it can take days or even weeks to make your unique plot files. Plotting uses a very slow hash known as Shabal. This is different from the SHA-256 hash used earlier in the article, which Bitcoin miners use rapidly. Since the Shabal hashes are hard to calculate, we precompute them and store them on a hard drive. This process is known as plotting your hard drive.

**Plotting:** Considering plotting, also known as generating a plot file, you create a number only used once (called nonces). Nonces are generated through consecutively hashing of data, including the account ID. The hard drive storage capacity allocated for plotting is directly proportional to the nonces that can be stored. One nonce ends up containing 8192 hashes. These 8192 hashes are organized in scoops, which are pairs of hashes. For each scoop a number is assigned between 0 and 4095.

During mining, a scoop number is calculated from 0 to 4095. Assume that the calculation gives a scoop number of 57. Then you will go to scoop 57 of nonce 1 and use that scoop information to calculate an amount of time, called a deadline. This process is repeated for each nonces that is stored in the hard drive. After calculating all of the deadlines, a minimum deadline will be picked. The deadline means that “the number of seconds that must elapse since the last block was forged before forging a block. If no one else has forged a block within this time-frame, the miner can forge a block and claim a block reward.”

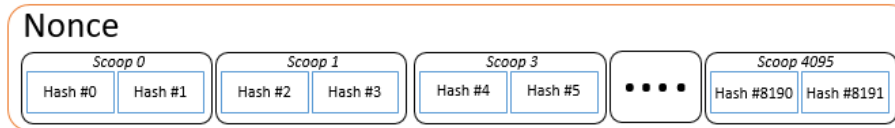


Figure 4: Plotting in BurstCoin: Nonces and Scops [17]

For instance, assume that a miner come up with a minimum deadline of 30 seconds then if no none of the miners could forge a block within the next 30 seconds, that miner will be given the opportunity to forge the block and get the reward for that block.



Burst [17] proposes PoC2 and PoC3 file structure where PoC2 is already underway as part of the Dymaxion effort. The primary change in PoC2 will be relegated to the formatting of the plot files in order to provide protection against the “time-memory tradeoffs” as outlined in the SpaceMint [14]. The authors of the SpaceMint researchers suggest that a miner doing a bit extra work could potentially have an edge on other miners while using less space. However, Burst PoC3 structure is considered a Post-Dymaxion change. It proposed to build on PoC2, but with the ability to store dual-use data (i.e. Burst plots and non-plot data).

In what follows we will have a look at some of the advantages and disadvantages of PoC.

#### 4.2.1 Advantages of PoC

- Proof of Capacity is Efficient. Using hard drives for mining is at least thirty times more efficient than using ASIC or FPGA based mining devices in terms of consumed energy.
- Proof of Capacity is Cheap. Any regular hard drive can be utilized for mining therefore other competitor miners do not gain any advantage from purchasing specialized equipment.
- Proof of Capacity is Distributed. Miners do not need to upgrade their device regularly since old hard drives can also store data as well as new ones. The mining device is reusable. As soon as a miner has completed mining, she can clear the hard drive and use it for some other purposes.

#### 4.2.2 Disadvantages of PoC

- The compatibility of PoC with blockweave approach is not straightforward. Different than PoA the nodes need to store all the data.
- Currently, the hard drives plot data that is useless beyond its mining purpose. However, there are plans to have the hard drives serve as redundant storage for important open source information. The hard drives could store maps, Wikipedia articles or other information worth preserving.
- Popular proof of capacity mining could lead to another arms race. Today people are using terabyte hard drives, but we could eventually see petabytes, exabytes, and zettabytes.
- There is already malware mining Bitcoin on people’s computers. If proof of capacity became popular, it’s possible you might see malware plotting people’s hard drives. The main difference is you are much less likely to notice some of your hard drive space being taken up.

Considering the above mentioned pros and cons, the most prominent advantages of PoC over the other consensus mechanism is the hardware usage that provide low energy consumption, cheap equipment, re-usability and do not require frequent upgrade. On the other hand, the compatibility of PoC with blockweave approach is not straightforward and different than PoA, the nodes need to store all the data. In our model, we combine PoC, PoS and PoA in a secure and efficient manner.

### 4.3 Proof of Stake (PoS)

Proof of Stake (PoS) is another category of algorithms that are commonly used to achieve consensus across a blockchain system. Strictly speaking, PoS is a mechanism that prevents

Sybil attacks (ie prevents a user from acting like  $N$  different person at the same time). In a PoS system, the vote power of a participant in a system depends directly on the number of stakes. This makes one with  $x$  coins cannot act as more than one person with  $x$  coins each. In order for a blockchain to live, new blocks must be created and added to the chain. Who has the right to create these new blocks? In a proof-of-work system, miners compete for this right by using computing power to solve random encryption puzzles. The winner starts by creating the next block and receives some rewards. The more computational power a miner has, the more likely he/she is able to publish the next block. On the contrary, PoS systems revolve around the idea of the amount coins a miner/block producer owns to publish the next block.

There are generally two variant of PoS algorithms:

- **Byzantine Failure Tolerance (BFT) Based Proof:** Instead of a random validator who is given the right to create a block that other participants have to accept, BFT systems represent the idea of bidding and accepting. As with a chain-based PoS system, a randomly selected validator (stack-weighted) can recommend a block to other validators. All examiners should communicate with each other until all honest examiners agree. If they reach an agreement, they accept the block and it will be the final block.
- **Chain-based Proof:** As with Bitcoin, a validator is randomly selected at each time slot to create a block that consists of the longest chain (longest chain). However, instead of choosing a validator based on who first solves cryptographic puzzles, the likelihood of selection is weighted based on the number of coins or 'stakes'.

#### 4.4 Proof of Access (PoA)

Proof of Access (PoA) is a new consensus mechanism introduced by Arweave project [22] that is built on the basis of PoW. A typical PoW system generates the next block only on the basis of the previous block, but the PoA algorithm combines the data of the previous block to randomly select a recall block. By getting the hash value of the current block and calculating its modulus relative to the height of the current block, we can choose to "recall block", which is to be merged into the next block. The advantage is that miners do not need to store all the blocks forming a blockchain.

The PoA consensus mechanism encourages the mines to get incentives by storing blocks. This mechanisms gives more incentivizes to mines by storing 'rare' blocks then storing well-replicated ones. This is due to the fact that when a rare block is preferred, miners with access to it compete amongst a smaller number of miners in the Proof of Work puzzle for the same level of reward. As a result, miners that chooses to store less frequent blocks on average gets a greater reward over time, all the others gets equal rewards. Therefore PoA performs a probabilistic and incentive-driven mechanism for ensuring the number of duplicated copies of any single piece of data in the network for reliability. On the other hand, different decentralised storage systems with different consensus mechanisms specify an exact number of redundant copies that should be provided for a given piece of data, and mediate this using a system of 'contracts' [4].

## 5 Consensus Algorithms of Stashware

Stashware is based on an combination of PoC, PoS and PoA as consensus mechanisms. iPoC is a new consensus mechanism in which PoS is integrated with PoC.

## 5.1 iPoC in Stashware

In Stashware, the consensus algorithm run between nodes is the improved version of proof of capacity (PoC) with proof of stake (PoS), abbreviated as iPoC. In this consensus, any node with staking at least multiple of 16,000 Stashware coin (SWR) can be a miner. A miner can be selected as validator for confirming the created block or can be creator to publish the block.

For confirming a block, 16 validators will be participated which are selected among 32 randomly chosen candidate validators. In this selection, the priority of each validator for confirming a new block will be determined by a secure *creator selection protocol* (CSP) that utilized a deterministic PRF function.

On the other hand, each block can be mined by any miners that owns least multiple of 16,000 Stashware coin (SWR). The chance of mining a block depends on the storage used on the computation of the puzzle for each block. Namely, in the block creation, proof of capacity works well.

We describe the four components of the Stashware consensus mechanism as follows.

- **Block creation:** In Stashware, participants who create blocks are called creators. Creators contribute their computational power to the network to validate transactions. For these operations, they are rewarded by the protocol in the form of freshly mined SWR. To be a candidate creator for a given block, a participant needs to own at least a stake of 16,000 SWR.

The miners uses their available hard drive space to decide the mining rights, instead of using the mining device's computing power. First of all, they plot their hard drive and the actual mining of the blocks. Depending on the size of their hard drive, it can take days or even weeks to make a unique plot files. In this phase, a very slow hash function known as Shabal will be used. For each block creation, they used their own plotted storage driver to compute the puzzle within a given minute (say 2.5 minute).

- **Block validation:** The miners owns at least 16,000 SWR can also be selected as validator. The more SWR a candidate has, the higher his/her chance of being given the rights to confirm the next block. For instance, if there are 200,000 SWR is activated for a block confirmation, and a validator with a stake of 80,000 SWR has a 40% chance of confirm the block.

Validators are ordered regarding their priorities resulted by the CSP. For example, if there are 32 candidate validators, the CSP could randomly select a priority list as follows:

- $Priority_1$  = Validator 3
- $Priority_2$  = Validator 2
- $Priority_3$  = Validator 13
- $Priority_4$  = Validator 19
- .
- .
- .
- $Priority_{16}$  = Validator 28
- ---
- $Priority_{21}$  = Validator 31
- $Priority_{22}$  = Validator 4

- .
- .
- .
- $Priority_{32} = \text{Validator } 15$

This means that the first twenty Validators (i.e, Validator 3, 2, ..., 28) will have the first priority in confirming the block. If Validator 3 does not confirm the created block within a given time interval (say 30 sec), one of the remaining Validator will replace that can compute within the time (e.g. starting from Validator 31).

The more SWR one owns, the greater one's chances of being given high priority. To confirm, a miner needs to put up a security deposit (your 'Proof of Stake') of 32 SWR per block created. This deposit is locked up for 4 cycles. This deposit can be slashed if the validator double confirms (the 'Nothing-at-Stake Problem').

- **Fork Choice Rule:** When two blocks are created at the same time, the block with highest number of validations and the longest chain will be followed.
- **Block Structure:** In traditional PoC, each block depends on the previous block. Stashware uses a new consensus mechanism that is based on PoC and PoA (see Section 5.2).

## 5.2 Proof of Access (PoA)

Now we briefly summarize the advantages of our PoC and (PoA) based consensus mechanism.

- Stasware uses a new consensus mechanism in which PoA is integrated with iPoC. However, Arweave is using PoA which is integrated with PoW. Therefore, our consensus mechanism is more favorable than PoA in terms of fast block generation, more data availability, more incentivization, efficient file auditing.
- PoA incorporates data from a randomly chosen previous a *recall block*, to create the next block. The recall block depends on a special algorithm which maximizes the detection of data loss.
- Recall blocks are chosen by taking the hash of the current block and calculating its modulus with respect to the current block height.
- Miners do not need to store all the blocks forming the blockchain but they are incentivised (by POA and wildfire) to store data of previous blocks in case it randomly gets pick as a recall block
- As the number of redundant copies of the block lowers, the incentive to keep that block increases - there will be less competition fighting over the next block if that copy of the block gets recalled.

## 6 Security and Privacy

Stashware provides the following security services

- confidentiality of key

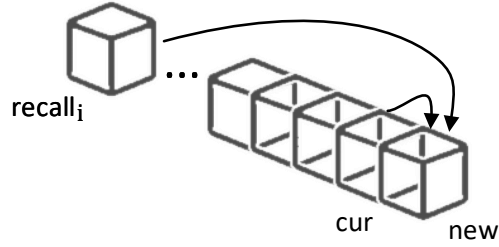


Figure 5: The sketch of blockweave structure.

- integrity of data
- compromise authentication
- compromise access control
- impersonation of existing users

We would like to highlight that the security and privacy requirements of each country depends on their own policies and regulatory compliance. For example, the USA has its own legislation for HIPAA (the Health Insurance Portability and Accountability Act) while EU countries have to follow GDPR (the General Data Protection Regulation). Stashware will provide any user friendly and transparent open source tools to ensure and fulfill all of the customers' security and privacy concerns.

## 6.1 Security and Privacy Services of Stashware

There is no confidentiality for traditional cloud storage providers since they have the full control of user data and fully access user files. This is certainly not appreciated by users and enterprises because of sensitivity of private data. Therefore, it is very crucial to store the data (as well as the metadata) in encrypted form as well as give the full control to the user.

Therefore, Stashware provides the state of the art efficient encryption methods (e.g., a signcryption method which is a public-key primitive that signs and encrypts the data simultaneously). The user has fully control of his/her encryption keys. Our encrypted has the following advantages with respect to its competitors:

- **Confidentiality:** Any external including internal adversaries (e.g., Stashware) could monitor the data moving in memory before it is stored in disk. Data (as well as the metadata) must be encrypted before it reaches Stashware and must always remain encrypted while in Stashware. Secure even if the end application or user accounts are breached.
- **Integrity:** The data must not be changed without the user consent. Prevent any unauthorized or malicious modification, manipulation or deletion. Malicious users cannot deny having signed the message.

- **Agile Cryptography:** Stashware will provide novel agile cryptography services which aim to meet current and future data security demands. Therefore, the Stashware platform will allow system flexibility and the ability to adapt quickly to newly developed cryptographic algorithms in an efficient and scalable manner.
- **Efficient key management:** The encryption key management should be designed in such a way that end users have shared control on the keys for data decryption.
- **Data at rest:** If the user data is sensitive then it should be encrypted on client side so that no one including Stashware can access the data without being detected.
- **Authenticity and Integrity:** Provide efficient signcryption methods (to provide both integrity and confidentiality of data as well as authenticity an efficient manner)
- **Signature aggregation:** The data will be split into chunks and will be signed by the user. In order to improve efficiency of Stashware, we aim to utilize signatures that allow signature aggregation (e.g., Boneh-Lynn-Shacham (BLS) [5] or Schnorr signatures [16]). Note that publicly verifiable homomorphic linear authenticators aggregate signatures into a single signature. There are two types of aggregate signatures: 1) aggregate and output a single compact (verifiable) signature given multiple individual signatures which have been signed with the same private key, 2) aggregate and output a single compact (verifiable) signature given multiple individual signatures which have been signed with the different private keys (i.e., a batch version for different users can also be aggregated.). This will allow Stashware to reduce the cost of communication (with short signatures) as well as computation of nodes.
- **Efficient hierarchical access control mechanisms.** The data can require to be a hierarchical structure (like a tree-like structure) such that certain data can only be retrieval for a specific attribute/role of a user.
- **Efficient data sharing mechanisms.** Encrypted data should be shared with anyone in an efficient manner (e.g, without being downloaded). Stashware provide an efficient mechanism using so called proxy re-encryption schemes which are used to transform ciphertexts without compromising any security and privacy issues under untrustworthy environments.
- **Security against offline attack:** Our security methods will provide semantic security which provides randomized encryption and prevents any types of offline attacks.

## 6.2 Secure Deduplication

The concept of secure deduplication is interesting for both research and industrial community because of reducing high costs in cloud computing environment (e.g., Amazon S3, Dropbox, OwnCloud, TeamDrive, Box, OneDrive (formerly SkyDrive), Google Drive, DepSky, and SugarSync). The first solution is to apply *convergent encryption* mechanism which is designed by Douceur et al. [9]. The data here is encrypted using a symmetric encryption scheme with a key which is deterministically derived from the hash of the data content. The convergent encryption mechanism is actively used by commercial CSPs like Amazon S3, Dropbox, Google, and Bitcasa [3, 18]. Note that convergent encryption does not provide semantic security because of content-guessing attacks using the deterministic nature of the content hashing <sup>1</sup>.

---

<sup>1</sup>Semantic security in this context means that ciphertexts leak no information about the underlying plaintexts except the knowledge of their equalities.

In public blockchain, the data can be plain or encrypted. Therefore, possession of an existing data on the chain can be achieved in two cases.

- In case of data is plain, any user can claim to posses the data on the chain. This is simply cloning data into repository of the new repository.
- In case of encrypted, the user who wants to posses the data on the chain should provide a proof of the data on the chain.

### 6.3 Public Auditing

Public auditing of public data will be provided with incentive. The integrity of a data is simply done through Merkle Hash Tree structure (MHT), which is a verifiable data structure. Each file or the collection of files will be divided into smaller blocks with a fixed size. Then, MHT is built by inputting the hash values of each data block  $F = (N_1, N_2, \dots, N_k)$  labeled as leaf nodes. After that, the internal nodes are computed by simply computing the hash value of its two child nodes with a cryptographic strong collision hash function  $h$ , such as  $h_d = h(h(N_i) || h(N_j)) \forall i, j$ . Suppose that the verifier keeps the root hash  $h_r$ , to check the integrity of  $N_3$  and  $N_6$ . The prover should provide only the  $\Omega = \{N_3, N_6, h(N_4), h(N_5), h_c, h_f\}$ . Then, the verifier computes a new root node value  $h'_r$  by reconstructing the MHT and checks whether  $h'_r$  is equal to  $h_r$ .

It will be optionally to deploy smart contract for auditing a file on the chain. In the contract, the frequency of file auditing, the incentive for each successful auditing, and other required parameters will be defined. Whenever, any full node in the chain can run the contract to audit file and get the incentive.

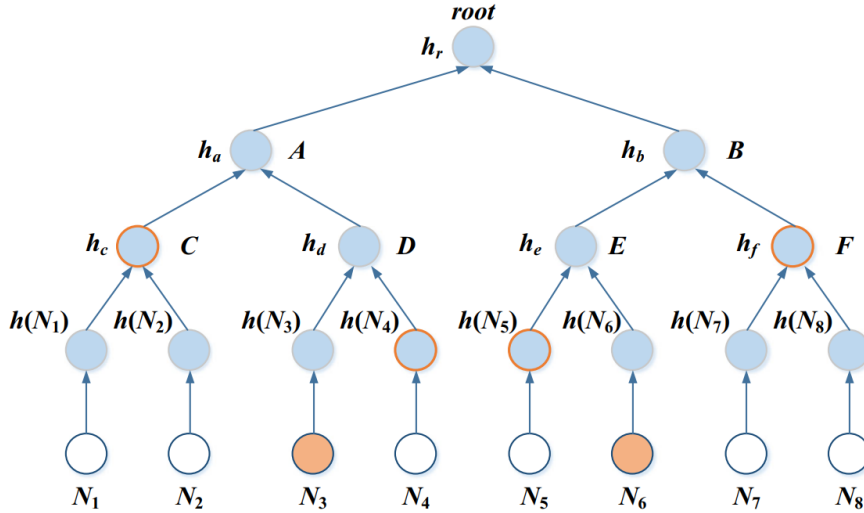


Figure 6: An example of Merkle Tree

### 6.4 A Privacy-Preserving Public Auditing

Privacy-preserving public auditing of a private data (which belongs to user or enterprises) will also be provided with an extra incentive on blockchain [21, 12, 19]. This is crucial for archived files to ensure that data remains on Stashware for a long time (e.g., 100 years). In Stashware, we will utilize existing constructions which have been designed for classical cloud infrastructures

and integrate with blockchain and allow anyone to be an auditor with incentive. On a very high level, the public auditing without compromising the privacy of data will be performed as follows:

1.  $i$ -th user  $U_i$  is delegating the auditing mechanism to a Trusted Third-Party Auditor,
2. a protocol for oblivious pseudorandom random function is run between the  $i$ -th user  $U_i$ ,
3. the  $i$ -th user  $U_i$  securely communicates with the Stashware,
4. the  $i$ -th user  $U_i$  uploads a file to the Stashware which supports both the secure deduplication and the privacy-preserving public auditing properties,
5. a privacy preserving public auditing is run between a third party auditor and the Stashware.

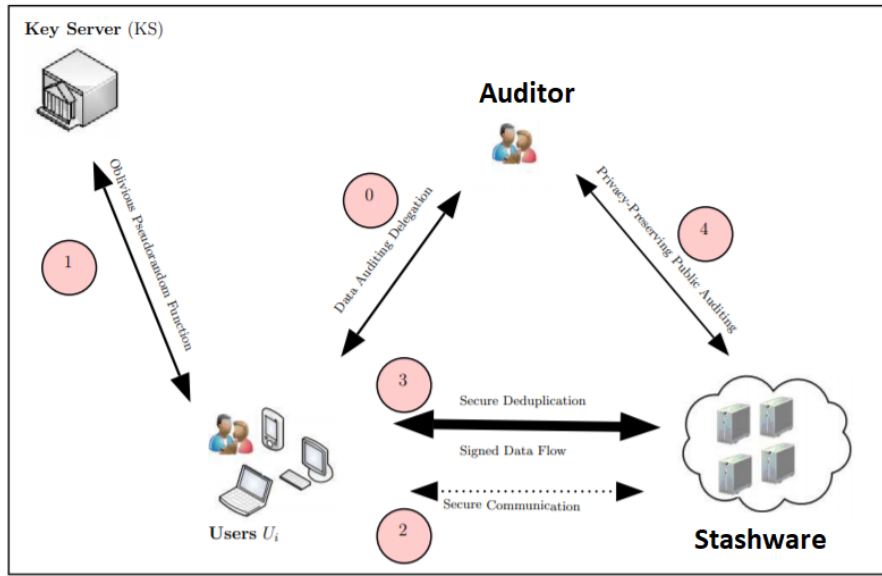


Figure 7: A Secure Deduplication Architecture with Stashware.

## 7 Related Work

In this section, the recent and popular existing decentralized storage based blockchain solutions are briefly described, such as Siacoin, Storj, Filecoin, Lightstreams and Arweave. This section also extensively evaluate Arweave solution by discussing its innovations, bottlenecks and weaknesses.

### 7.1 Siacoin

Siacoin is a blockchain-based decentralized storage platform. It is introduced by a Nebulous Inc company. Siacoin stores the data by dividing into some fragments then perform cryptographic



encryption for distributing it across Sia decentralized network. Sia project uses proof of storage as a consensus mechanism.

The target of Siacoin is to reducing the overhead costs of cloud storages significantly by allowing users to host or in a sense “rent out” their free/unused storages and hard drives. Therefore many people are calling this project as “Airbnb of hard drives”. By offering low-cost and decentralized storage solutions, Sia project has set out to disrupt some big players in the industry such as Amazon A3, Onedrive, Dropbox and Google Drive.

Some of limitations of Siacoin project are (i) difficult to scale and handle large data volumes due to blockchain limitations (ii) requires the download the entire blockchain first in order to become a farmer, which requires allot of space and hours of synchronization. (iii) requires a deposit of SIA coin before-hand in order to rent space on the network, (iv) only allows the payment with the native SIA cryptocurrency (v) lower ease-of use for both traditional companies and individuals.

## 7.2 Storj

Storj is another decentralized storage project which is built on the Ethereum network. This project has quite many community users and is dedicated to the open-source and user-experience ethos. Storj provides a platform, cryptocurrency, and suite of decentralized applications that provides customers to keep data in a secure and decentralized manner. Storj uses the Ethereum public chain to achieve the consensus, without its own consensus algorithm.

Storj’s storage technology involves file sharing, similarly to torrents working approach and separates parts of the files to users in the network. User’s files are encrypted before uploading. Customers has their own private key to validate the ownership. As soon as a customer wants retrieve his/her file, they request it and Storj uses a distributed hash tables to locate all the shards and piece them together.

Some disadvantages of Storj project are (i) no storage marketplace for dynamic pricing, (ii) single point of failure of the Storj service through storj bridges, (iii) more subjected to vulnerabilities (like dDOS attacks) due to the fragmented solution.

## 7.3 Filecoin

Filecoin is an open-source, cryptocurrency and decentralized storage system. It aims to be a blockchain-based cooperative digital storage and data retrieval method that targeting to replace HTTP. This project is owned by Protocol Labs and builds on top of IPFS, allowing users to rent their free and hard drive devices. However, it is not capable for permissioning access to protected content.

The Filecoin DSN protocol can be implemented on top of any consensus protocol that allows for verification of the Filecoin’s proofs. It allows periodically producing cryptographic proofs ensuring that they are providing the required capacity. In Filecoin, not only are miners given stash rewards for hosting files, but they must prove that they are continuously replicating the files for more secure storage. They are also rewarded for distributing content quickly — the miner that can do this the fastest ends up with the coins. One of the disadvantages of Filecoin is customers have to pay for storage and retrieval of their own file. Also, Filecoin is lack of an incentivization layer that can help in its mass adoption of modern decentralized storage technology. Although it plans to add smart contract capabilities in the future there is no indication that this will be Ethereum-based, thus losing the benefit of access to the largest community of developers and tools for smart contracts.

**Some problems of Filecoin that need to be solved are as follows:**

- A technique and proof of evidence is needed to confirm the customers data has been deleted by the nodes (storage miners) after data retrieval so only customers have access to it from that point on.
- Bridges or connections that operate outside of the blockchain and allow communication between different blockchain technologies will have to be invented so that smart contracts can function on the Filecoin blockchain.
- How to verify the security and privacy of stored data and ensure the nodes has not retrieved by decrypting the customer's confidential data during storage.
- There is no clear mechanism that storage providers and storage retrievers will be efficiently rewarded and incentivized that they try to protect their reputation and not to act in a malicious way.

Comparing Filecoin with other decentralized storage such as Arweave: Arweave's protocol design has an embedded mathematical pricing function, load-balancing demand and supply at fair prices. Any file can now be published anonymously to the world, and stored forever without powerful third parties able to take them down. However, in file coin there is still no clear mechanism for storage providers that encourage them to be efficiently rewarded or incentivized. Depending on the lifetime of storage, the price for the Arweave tokens is cheaper than monthly fees for competitors such as Filecoin. Also, Arweave's structure created more scalable infrastructure to be the backbone for the data economy. This is because of the fact that Arweave's structure is able to reach up to 5000 transactions per second (TPS), and store data at a fraction of the cost of Filecoin, Siacoin, and Storj.

In overall, Filecoin will have to compete with current cloud storage providers and that may be a tall order. They will have to meet or exceed competitors on speed, security and reliability. Finding novel solutions to the latency problems that plague any peer-to-peer system will be critical to customer adoption.

## 7.4 Lightstreams

Lightchain is an Ethereum-compatible blockchain which aims to provide privacy-enabled, content-sharing (movies, music, documents, blogs, posts, and other digital assets etc) on blockchain. Lightstreams provides its own MainNet network, and a testnet called sirius. It is a permissioned decentralized storage system and uses proof of authority as consensus mechanism that restrains exchange from having a lot of nodes. It is also based on IPFS for uncapped and free data storage. It provides Ethereum for smart contracts and Tendermint infrastructure for blockchain Proof of Authority. Private and large sized files are stored decentrally using the IPFS protocol with an additional security layer for protecting access to content.

## 7.5 Arweave

Arweave is a blockchain storage project similar to IPFS/FileCoin. Its vision is to solve the problems of the Internet, where it believes that data can be easily changed, regulated and lost. Arweave hopes to be a complementary system to the Internet, using the blockchain approach to permanently store data, which cannot be changed, and using token incentives to achieve sustainability. It proposes a number of technical innovations to address existing blockchain issues, including: BHL, WL, BlockWeave, Proof of Access, Wildfire, BlocksShadows. These technical points complement each other to form a complete and reasonable solution.

In general, Arweave proposes a number of solutions to the current blockchain problem, and deeply contemplate the use of token incentive mechanism to solve it, thus achieving a dynamic balance. For example, if each node does not need to store all blocks of data, what data does it need to store? Arweave's incentive mechanism leads people to store "rare blocks" so that they have a higher probability of getting blocks out. That's where the economic incentive comes in. The wildfire rating system also uses tokens to motivate users to respond to requests and thus make the overall system better. It's great from a conceptual point of view, and it hits a lot of the sore points of the current chain.

### 7.5.1 Weaknesses of Arweave

The actual application scenario is too narrow for developers to use, and the Arweave feature can be applied to HTML5 web pages. According to the security, establish decentralized H5-APP. Arweave's data are all stored on blockchain. In practice, however, we have a narrower set of scenarios for issuing this depository. As you can see, the most stored in Arweave at the moment are screen-shots of some of the anti-government comments from Twitter. At the same time, Arweave's feature is that you can never tamper with it. This is especially difficult when you're developing an application because if you upload it to Arweave, you have to make sure that there are no errors. If there are errors, even a single punctuation mark, the content you've uploaded will be invalidated and you will have to re-upload it, which will result in a lot of garbage. In addition, due to the openness of blockchain, the content posted on Arweave is open to the whole society, which is not suitable for uploading personal content.

Although the team says Arweave is IPFS-compatible developers building on IPFS can seamlessly transition to Arweave. However, if developers develop based on Arweave, they will not be able to directly update their HTML5 apps. Developers must simply abandon the old version, and there is bound to be some inconvenience in re-uploading the new version.

The business model is relatively simple, which may trigger a price war for homogenized projects. Arweave focuses on one-time payment and access to permanent file storage. This model is relatively simple, but there is a risk that it will lead to homogenized projects that use the same storage concept and start a price war.

In addition, as mentioned at the beginning, it requires the integration of so many new technologies to form a complete and reasonable solution. Various technical points need to be supported and supplemented each other, and it may not be of great significance to draw out a single point for reference. In the current situation, its token mechanism has been preliminarily able to run the version, but its actual business scenario application function. On June 15, 2020, the first storage function directly over 80KB large files was available, all with test code. So the value of its practical application is still far away.

Recent code has also been desperately trying to fix various file storage synchronization bugs. This is basically its main function, and if it doesn't have an OK, it's actually of very little value. And above do application development, it is basically impossible, its intelligent contract application development with JS API is a complete simulation demo. There were only 2 small JS files, and they were not updated for 7 months. Recently, they were suddenly updated for 5 days. All of them were warning and annotated.

## 8 Development

The project will be implemented Golang.

## 9 Future Work

Our experience designing Stashware revealed several problems which have no simple solutions; further research is required.

- **Data in process:** Data needs to have complete obfuscation/encryption when processing data in memory. Secure Multiparty Computation (SMPC) allows performing arbitrary computation on the blockchain without having to download and decrypt [15, 7].
- **Password-protected secret-sharing (PPSS):** PPSS schemes eliminates to hold hardware tokens but only passwords achieving semantic security [1, 11, 2, 6]. This significantly improves the usability of Stashware as users can use any device without possessing hardware devices/tokens.
- **Private Search:** Stashware aim to provide a private search mechanism on an encrypted decentralized storage. Although decentralization eliminates the traditional data loss of the classical cloud providers it is still not possible to search on ciphertexts. Therefore, the only solution is to decrypt the ciphertexts on the nodes which allows privacy leakage of the data owner. Private search mechanisms (e.g., oblivious keyword search) can be utilized to mitigate this issue [8]. Namely, private search allows a user to retrieve the data associated with a chosen keyword in an oblivious way.

## References

- [1] Michel Abdalla, Mario Cornejo, Anca Nitulescu, and David Pointcheval. Robust password-protected secret sharing. In *Computer Security - ESORICS 2016 - 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 26-30, 2016, Proceedings, Part II*, volume 9879 of *Lecture Notes in Computer Science*, pages 61–79. Springer, 2016.
- [2] Ali Bagherzandi, Stanislaw Jarecki, Nitesh Saxena, and Yanbin Lu. Password-protected secret sharing. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, pages 433–444, 2011.
- [3] Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart. Dupless: Server-aided encryption for deduplicated storage. In *Proceedings of the 22Nd USENIX Conference on Security, SEC’13*, pages 179–194. USENIX Association, 2013.
- [4] Juan Benet. IPFS - content addressed, versioned, P2P file system. *CoRR*, abs/1407.3561, 2014.
- [5] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 514–532, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
- [6] Jan Camenisch, Anja Lehmann, Anna Lysyanskaya, and Gregory Neven. Memento: How to reconstruct your secrets from a single password in a hostile environment. In *Advances in Cryptology – CRYPTO 2014*, pages 256–275, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
- [7] Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

- [8] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. In *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS '06*, page 79–88. Association for Computing Machinery, 2006.
- [9] J.R. Douceur, A. Adya, W.J. Bolosky, P. Simon, and M. Theimer. Reclaiming space from duplicate files in a serverless distributed file system. In *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*, pages 617–624, 2002.
- [10] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. pages 139–147. Springer-Verlag, 1992.
- [11] Stanislaw Jarecki, Aggelos Kiayias, Hugo Krawczyk, and Jiayu Xu. Highly-efficient and composable password-protected secret sharing (or: How to protect your bitcoin wallet online). In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany*, pages 276–291, 2016.
- [12] M. S. Kiraz, İ. Sertkaya, and O. Uzunkol. An efficient id-based message recoverable privacy-preserving auditing scheme. In *2015 13th Annual Conference on Privacy, Security and Trust (PST)*, pages 117–124, 2015.
- [13] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.
- [14] Sunoo Park, Albert Kwon, Georg Fuchsbaauer, Peter Gazi, Joel Alwen, and Krzysztof Pietrzak. Spacemint: A cryptocurrency based on proofs of space. Cryptology ePrint Archive, Report 2015/528, 2015. <https://eprint.iacr.org/2015/528>.
- [15] Benny Pinkas, Thomas Schneider, Nigel P. Smart, and Stephen C. Williams. Secure two-party computation is practical. In *Advances in Cryptology – ASIACRYPT 2009*, pages 250–267, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [16] C. P. Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology — CRYPTO’ 89 Proceedings*, pages 239–252, New York, NY, 1990. Springer New York.
- [17] Robert Stadler Seán Gault, Franz von Ancoina. The burst dymaxion: an arbitrary scalable, energy efficient and anonymous transaction network based on colored tangles, 2017.
- [18] Jan Stanek, Alessandro Sorniotti, Elli Androulaki, and Lukas Kencl. *A Secure Data Deduplication Scheme for Cloud Storage*, pages 99–118. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
- [19] Choon Beng Tan, Mohd Hanafi Ahmad Hijazi, Yuto Lim, and Abdullah Gani. A survey on proof of retrievability for cloud data integrity and availability: Cloud storage state-of-the-art, issues, solutions and future trends. *Journal of Network and Computer Applications*, 110:75 – 86, 2018.
- [20] Vivek Vishnumurthy, Sangeeth Chandrakumar, Sangeeth Ch, and Emin Gün Sirer. Karma: A secure economic framework for peer-to-peer resource sharing, 2003.
- [21] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou. Privacy-preserving public auditing for secure cloud storage. *IEEE Transactions on Computers*, 62(2):362–375, 2013.
- [22] Sam Williams, Viktor Diordiiev, Lev Berman, India Raybould, and Ivan Uemlianin. Arweave: A protocol for economically sustainable information permanence. 2019.