# BitcoinHD1

*The Crypto Currency System Based on CPoS*

BitcoinHD1 Core Team

June 7, 2023

# Contents

# Chapter 1

# BHD1 Introduction

BHD1 is a new crypto currency based on the CPoS(Conditioned Proof of Space) mechanism. By using hard disk as a consensus participant, it can significantly lower energy consumption and entry barrier, making mining of crypto currency safer, more decentralized and for everyone. BHD1 generates its unique value through mathematics and code. This White Paper will explain and elaborate on the monetary and technical attributes of BHD1.

## 1.1 Crypto Currency

When it comes to crypto currency, before the well-known Bitcoin, the entire crypto community has begun to experiment on a better international payment channel, such as Dai-Wei's Ripple and B-Money.

Ripple has been used in the settlement between banks in different countries, but never became quite as popular as was Bitcoin, because it is considered too centralized for a crypto currency. Compared to those decentralized crypto currencies, Ripple has always been more appealing to enterprise and business users, but less to the crypto enthusiasts, because its token generation procedure does not involve or incentivize the crypto enthusiasts.

B-Money causes network congestion due to the need for network synchronization in its design. At that time, the network speed was not so fast. During the sending and receiving of currency, network lag often caused problems, sometimes user receives no reply while waiting for a network packet. The system was impractical for mass adoption.

Then Bitcoin came to stage with its own Nakamoto consensus, which is the asynchronous PoW consensus. In the early days, no one was optimistic

about this project. The consensus did not use simultaneous transactions to ensure that transaction results are right, but instead adopted a very interesting mechanism: the longest chain. That is to say, in this distributed system, the nodes manipulates packages and composes the chain, which includes the transaction with the correct result. For this specific package to appear, of course, the nodes in this system have to jointly verify. Only given a timeout package and only when more people participate in the accreditation before timeout, will this package reach consensus.

In this system, there is a situation where nodes can collectively do bad things, so that the correct transaction is not packaged, and the transmission of the network is invalid. Since asynchronous system avoids excessive communication in the network, it is more suitable for multiple-step transactions, While the risk of this mechanism lies with the possibility of the majority of CPU power being controlled by dishonest nodes. A good example is the later appeared 51% double spend attack. The last thing the financial system should do is to roll back or double spend, that is also why Bitcoin was not widely accepted at the beginning.

Over time, a lot of participants joined the system for the financial benefits. Since the difficulty (for manipulating package mentioned above) of the system raised, the cost of harassing the working system has greatly increased for the bad guys. More stable the system, more profitable being honest rather than being dishonest. At this time, people began to realize the fascination of this crypto currency and numerous fans appeared. After years of difficulty increase, the BTC system has gradually stabilized, making it much harder to do double-spend or rollback. It also inspired the original teachings of Bitcoin, and gathered many crypto enthusiasts. It also inspired the original teachings of Bitcoin, and gathered many crypto enthusiasts. At this time, several new types of crypto currencies had been born or made by forks and copies, many got attacked because of their low computational power. The systems with low difficulty are unsafe and can be easily attacked, while highly available systems need tremendous energy consumption.

We can put a summary on Bitcoin tech features below.

Bitcoin was never aggressive on using new tech, but chose to adopt relatively mature technologies to build a safe and reliable Peer-to-Peer cash system.The more validated and simple the technology is, the more secure and trustworthy the system will be. For example, the SHA256 algorithm in

Nakamoto consensus, is designed by NSA (US National Security Agency), with proven reliability. It seems that the initial design never considered the current ASIC (Application-Specific Integrated Circuit) and power monopoly issues, but focused on pursuing ultimate system security, even sacrificed some of the high efficiency or high concurrency features of internet.

## 1.2 Seeking Alternatives

When numerous resources are being used in the mining procedure and costs are gradually increasing, crypto currency enthusiasts have started looking for alternatives to lower power consumption in two different ways: either using new consensus to lower energy cost or using more general apparatus to lower the cost of mass production. The golden age of ASIC mining device and anti-ASIC algorithm implementation had come. The original intention of Ethereum and Monero was to resist ASIC, using a different non-ASIC-friendly consensus to keep the system away from ASIC manufacturers' manipulation while keeping the energy consumption low. However after a period of time, ASIC manufacturers still found ways to design devices that would work with the corresponding algorithm. Among those ASIC ones, Litecoin has to be mentioned. It started with Scrypt which is an anti-ASIC mining algorithm, and soon ASIC manufacturers started producing ASIC mining devices that could work with Scrypt.

BHD1 provides the perfect solution for the issues mentioned above. It brings a method for crypto zealot to make general apparatus while keeping the energy consumption low. Meanwhile, BHD1 maintains a relatively high difficulty level to ensure the stability of the system by using its consensus Proof of Space (abbr. PoS). The PoS consensus used by BHD1 is also one of the most decentralized consensus mechanisms in this era. Compared with the PoW, where hash power rules, the PoS consensus is ruled by storage power, but slightly different from the cloud storage. PoS utilizes hard disks as a more economical consensus method, so that more people can participate in construction of the system-stabilizing hash power with their own devices. It was the original intention of Nakamoto to design PoW, a decentralized system and an innovative path to real decentralization for everyone, raising consciousness in every new comer to think about and overturn the existing design. BHD1 has inherited BTC's spirit, now the new PoS mechanism is responsible for bringing a better future for crypto currency, and engaging more people in the construction of the economic system.

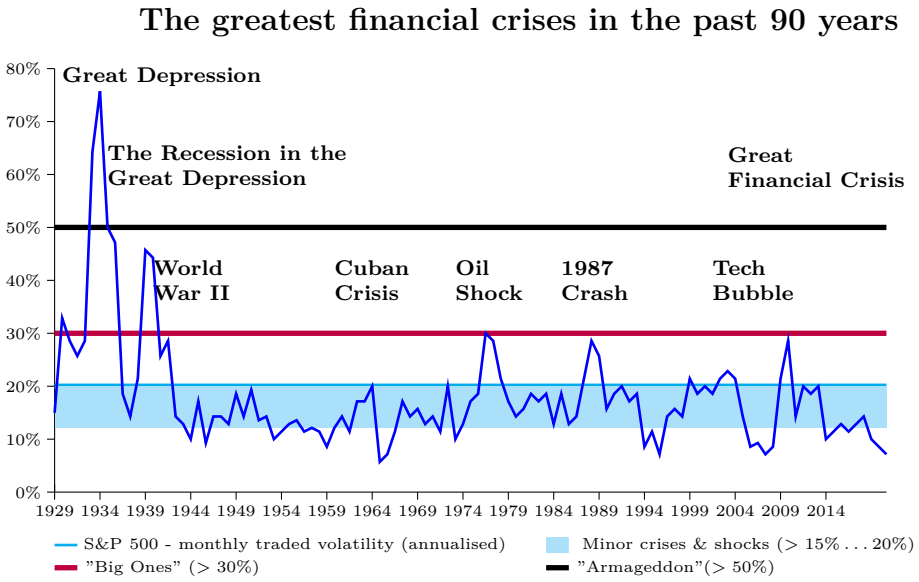
# Chapter 2

# The Four Major Problems

Monopoly, centralization of computing power, high energy consumption, and the incompleteness of existing PoSs have become the four major problems in the Crypto industry. From the beginning of its design, BHD1 is aimed at solving the four major problems.

## 2.1 Monopoly

Since its inception, Bitcoin has always had the mission to solve financial institutions' crisis of confidence and issue of monopoly. Since the financial crisis in 2008, Nakamoto believed that the centralization of the financial system would lead to repetition of the history, thus decentralization could be an effective solution for the economy.



So after all these years, what is the current status of Bitcoin?

The figure below shows value curve of the entire cryptocurrency market led by Bitcoin. Does it not look like fluctuations in the financial crisis cycle? Which brings us to think if Bitcoin is still decentralized.

The technology of Bitcoin-core is controlled by the core developers, and the code update speed is very slow, which can be described as code-centralization. Bitcoin's computing power is tremendous, ordinary people and personal computers cannot take part and can only trade in exchanges, which indicates hash-power-centralization. Bitcoin's block generation time is relatively slow, about 10 minutes per block, single digit TPS (Transaction per second) cannot provide the same experience as the current internet. Bitcoin core wallet did not make any UI/UX enhancement in ten years, and there is neither a core mobile version, nor any consideration of the current user experience, which indicates user-experience-centralization. Some people are planning to deploy lightning network, allowing more centralized companies to join the nodes, turning the Bitcoin system into a centralized payment system with far worse user experience than visa.

Many believes that the existing Bitcoin system needs to be changed or overturned, keeping the decentralization spirit and involve everyone in this revolution. BHD1 has a more economical decentralization approach, using lower cost storage instead of CPU/GPU power. If we believe that centralization can cause crisis to reappear, then we need to know that monopoly has to be eliminated to avoid any risk of potential crisis in the crypto world.

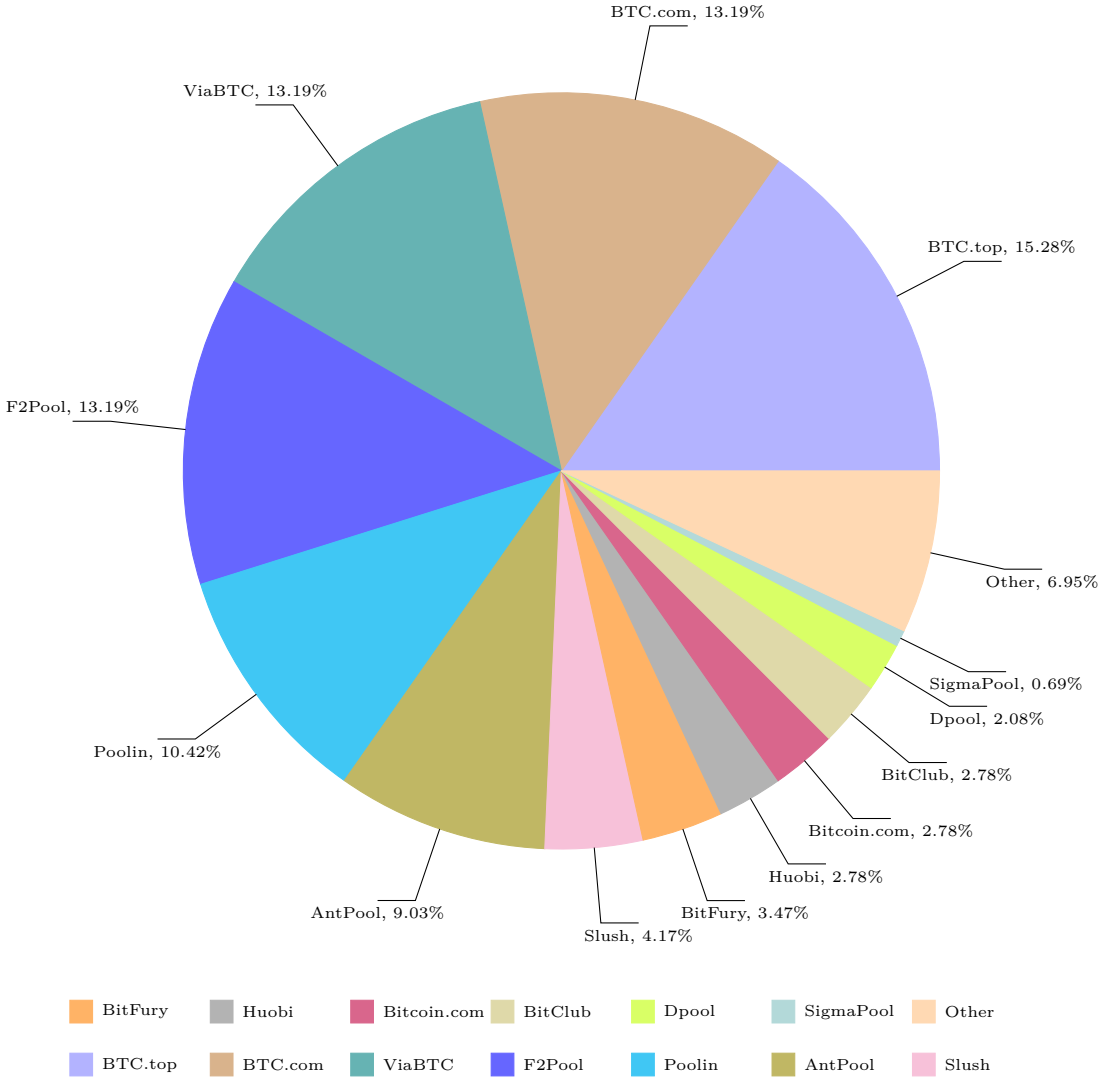
## 2.2 Power Centralization

We mentioned, the main reason for Bitcoin to prevail and be successfully used as the digital money is that its hash power has been maintained at a relatively high level. In 2017, Bitcoin hash power was 4400P, daily production was 1,800 coins, every peta hash power generated 0.4 Bitcoin on average, and consisted of 166 units of 6 tera hash power mining machine. Here comes the issue, the price of Bitcoin can be influenced by mining machine manufacturers through adjusting the price of the mining machine. Thus as crypto currency participants anticipate an increase in Bitcoin's earnings, everyone is willing to mine with higher hash power machines, and enjoy a higher possibility to get rewards through packaging. The top four

companies in Bitcoin mining account for about 53% of the mining share; The Ethereum system has a higher concentration ratio, the top three mining agencies account for 61% of the mining share. In addition, 56% of the world's Bitcoin mining software and 28% of Ethereum mining software are concentrated in the data center, showing that Bitcoin's operations are more corporatized.

The figure below shows that now Bitcoin's hash power is about 30,000P - 40,000P. Compared to year 2017, the hash power has increased by 10 times, which means the difficulty has also increased by 10 times for participants.

As shown in the figure below, the hash power has begun to be corporatized, or organized as pools nowadays, e.g. F2Pool, AntPool, Slush.



As hash power gradually increased, the mining machine manufacturers raised the difficulty of coin generation by making devices with improved configuration, kicking out many out-of-date device holders and discouraging a large amount of new entrant.

BHD1 solves the problem by using hard disk related consensus to disperse centralized hash power. In the existing PoW crypto currency, each collision of the hash value requires a large amount of calculation, which is of course also a method of managing difficulty. BHD1 writes the results of each collision on the hard disk in a pre-computed way. This is also a common time-for-space method to reconstruct the entire calculation. That is to say, under different block difficulties, it takes time and calculation, which takes power consumption differently. While in the BHD1 system, as long as the hard disk has enough storage space to contain a sufficient amount of answers the system can involve every crypto enthusiast in the block generation process, without the need for repeated large amount of calculations.

Bitcoin block generation process is roughly like the scrabble game, combining hints with given characters to form a complete word. It is hard for beginners to figure out what the word is, and not easy or at least time consuming even for the veterans. Comparatively, BHD1 is more like using a search engine, e.g. Google, to find the word, since the results are all pre-calculated. So the more words in the database, the higher the possibility to get the result. Compared with Bitcoin, BHD1 has a much lower barrier to entry, and is much more accessible for every individual.

The problem of hash power centralization can be resolved through such a space-for-time approach. Of course, this is just one of the problems BHD1 targets.

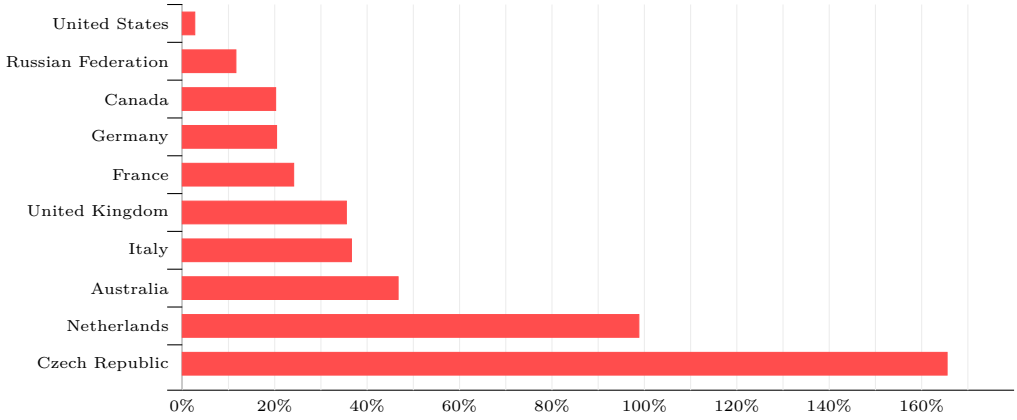
## 2.3 Energy Consumption

The concentration of hash power also brings about the problem of high energy consumption. So how much resources does the specific calculation consume?

To give an example, the current energy usage level of Bitcoin is enough to generate electricity for 10% of Italy, as shown by the figure below. That is to say, the resources used by Bitcoin could meet the needs of Rome, Milan and Venice, with a combined population of 6 million. Just as a popular

saying all roads goes to Rome, if the Bitcoin makes its way to Rome, it will also consume all of Rome's electricity.

**Bitcoin Energy Consumption Relative to Serveral Countries**




Since most miners are in mainland China (e.g. BitMain the famous manufacturer), I will give a Chinese example. Now that Bitcoin's hash power is around 45EHash/s, then in the case of 1 peta computing power and 0.1 yuan per kWh, it takes about 140000 kWh to do the specific calculation, costing an average of 14,000 yuan. China's high-speed railway consumes more than 9,600 kilowatt-hours per hour. For the 5 hour trip from Shanghai to Beijing, the train needs to use nearly 48,000 kWh. The current energy consumption of a Bitcoin is more than enough for a high-speed rail to run a return trip from Beijing to Shanghai.

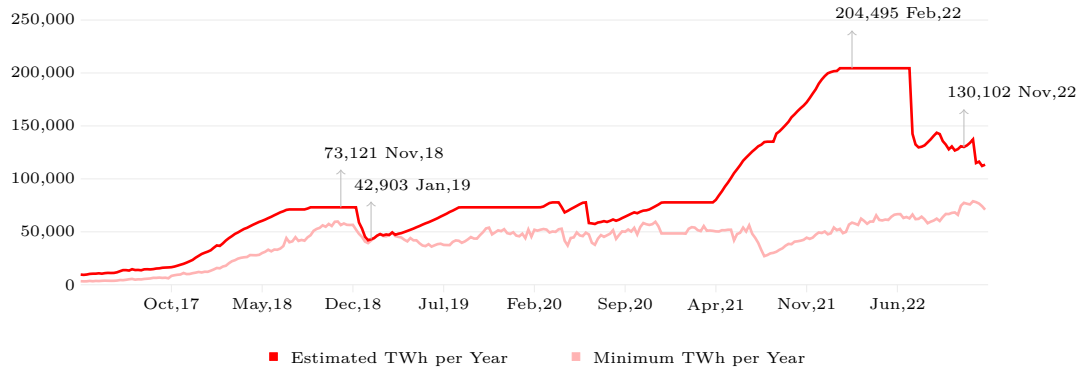
So what is the energy consumption of BHD1?

According to the comparison between a current second-hand S9 and a current second-hand 8 terabyte hard drive, the energy consumption ratio is about 1/300. That is, for 200 USD equivalent of electricity, ASIC takes 1700 watts, GPU takes 250 watts. Comparing those to 8 TB hard disk which cost around 200 USD, the hard disk takes only 5 watts. So for spending on 100 S9 or 100 8 TB hard disk devices with the same total amount of money, the S9 ones consume 122,400 kWh monthly, while for mining BHD1 hard disk ones consume 360 kWh, which is equal to only 5 days of electricity for an average American family. That makes BHD1 more accessible for anyone interested to participate and contribute in the long term. With such huge difference, the energy saved can be spent on more entities rather than on repeated consumption. Unlike Bitcoin which has slowly become a game for only few, BHD1's low power requirement keeps its door open for many.

Another energy related issue is even more serious: PoW's hash power is reflected in energy consumption, but energy is controlled by the national government in most countries, hence with the expansion of hash power, the impact of energy will gradually increase.

It is shown in the figure below, the energy consumption of Bitcoin was 73,121 TWh in October 2018, and decreased drastically to 44,722 TWh in January 2019. This fall in computing power caused by energy reduction affected the difficulty level of the entire block and the profit of mining machines. This disaster did not only hit Bitcoin. The minor crypto currencies with PoW consensus had to take the risk of forking. It was a deadly threat to the correctness and validity of the entire consensus.

**Bitcoin Energy Consumption 2017-2022**




That is to say, if the mining pool is concentrated under any centralized institution, then the institution can influence the system's difficulty level and benefits by adjusting the relevant energy resources. The computing power would drop dramatically due to a potential large-scale electricity power decline, which could even cause a PoW coin to fork. The low power consumption of BHD1 also provides an effective solution to this problem, through reducing the dependence on energy and taking a block generation approach that is more suitable for long-term survival. The PoS consensus is a low-energy-cost alternative to the current high-energy-cost ASIC based ones. By using the whole global hard disk storage as medium, PoS generates random numbers to guarantee high level of security, and ensures stability of the blockchain infrastructure.

## 2.4 Existing PoS Currency Design Issues

Has anyone considered designing crypto currencies using the designs of hard disks before? The answer is yes, there was Burst in 2014. Burst quickly

promoted the PoS mechanism and gained a lot of supporters, but at the same time exposed some of the inherent issues about the original PoS mechanisms. With those in mind, there has been a series of changes in the BHD1 tech structure.

At the beginning of Burst's design, there lacked a proper incentive mechanism. A huge part of the coins was mined by the supporters who joined earlier at a very low cost. Without the team's promotion effort, the participants who entered Burst later lacked motivation, and slowly this PoS currency faded out of sight. BHD1 adopts a dual incentive approach, mining could be done by staking or non-staking to balance operation team's costs and miner's benefits: miners can get all the benefits when they are staking at designated ratio; When the miners are withdrawing, the rewards are distributed to the operation team. BHD1 uses the conditional approach to ensure sustainable development of the chain and continuous introduction of new miners to maintain a long-term positive community development.

## 2.5 Why Does BHD1 Appear Now?

It is the existence of the above four problems that made BHD1 come into being.

As the number of crypto enthusiasts increases, the idea of decentralization has gotten bigger. Of course, everyone who is in the industry wants to be able to benefit. As Bitcoin's energy consumption is increasing each day, mining machine manufacturers are becoming more centralized. The crypto currencies based on PoS is in more demand now than ever. In addition, PoS consensus mechanism guarantees that the difficulty level can be quickly controlled, accumulate enough to maintain the normal operation of the system, and reward the transactions. BHD1 is superior to the existing crypto currencies in all the above areas. Its technology has been improved on the basis of Burst, and completely surpasses many other crypto currencies in technical and community dimensions.

Compared to the overhead energy assurance algorithm, we believe that low power consumption can also give the algorithm enough credit to ensure that everyone can use crypto currency in the future in more scenarios.

# Chapter 3

# BHD1's Technical Solution

BHD1 uses PoS as the basis of its consensus mechanism, it ensures sound development of the entire crypto currency by designing a long-term incentive economic model. At the same time, it has made some improvements to the existing PoS and upgraded it to the CPoS consensus.

## 3.1 BHD1 Distribution and Mining Mechanism

*Please note: some of the chain parameters are updated after BHD1 upgrade to Chia's consensus. Check the chapter "Update to Chia consensus".*

| | |
|---|---|
| **Total supply** | 21 million |
| **Development team** | 10%: 2.1 million. Way: pre-mined |
| **Operation team** | 5%: 1.05 million. Way: obtain from blocks generated during miners mining |
| **Miner** | 85%: 17.85 million. Way: mining |
| **Avg. block time** | 3 minutes |
| **Initial block size** | 15 BHD1 / Block, 2MB block size |
| **Halve period** | In 4 years, the first halving time is about 582688 block height |
| **Current TPS** | 70 transactions / sec |
| **Stake** | 1T hard disk stake 3 BHD1. Note: 1T hard disk is evaluated based on computing power, not absolute values. |

In the first month, after mining of the genesis block, miner can mine with no condition limitation. From the second month onwards, the 30% return that does not meet the conditional capacity mining revenue remains the same. For the remaining 70%, 43% is directly accumulated to the next eligible conditional address, and the remaining 27% is allocated to the BHD1 Development Foundation. For every 33,600 blocks (about 10 weeks) after, the BHD1 Development Foundation's revenue will be reduced by 2%, and the partial accumulation will be reduced to the next block that meets the conditions for mining. By the first halving, the BHD1 Foundation's revenue will be reduced to 5% and stay at that amount. 65% of the income that does not meet the conditional capacity mining will accumulating to the next block that meets the conditional mining; If the conditions are satisfied, miner would get 95% of the mining reward, then the 5% remaining reward would go to the foundation for marketing.

Conditioned Proof of Space, or CPoS, would lead the miners, mining pools, the foundation and other participants to engage in a positive business cycle, so that the whole system would always have a dominant temporary commercial vested beneficiary (this vested beneficiary could change with variables such as time, price and mining difficulty) to promote the whole ecosystem.

## 3.2 BHD1 Economic Model

BHD1's economic model / consensus mechanism has been upgraded based on the Burst PoS2 (Proof of Space), and is called: CPoS (Conditioned-Proof of Space).

The model will solve the problems listed below:

### 3.2.1 Economic Model Attack

The main purpose of miners mining is the payback period, and the benefits will inevitably lead to the sale of all mining output, resulting in market crash, lower prices and thinner profits. The CPoS mining model binds miner to its ecosystem, and uses output of mining as future input of mining, to make the entire BHD1 system grow automatically.

### 3.2.2 POW High Maintenance Cost

It requires a huge amount of power to keep the chain with PoW consensus safe. In good days, it works fine for each part of the system, but in hard times, miners have to pay bills by selling, and it is not easy to keep the miners in the system, if they always have to consider how much energy has been consumed.

### 3.2.3 Lack of Long-Term Economic Incentive

Without operational incentive funds, the promotional efficiency and market confidence is low. even the core technology might fail to get continuous update. As a result, effective development and iterations are non-existent in the long-run, the team may even create a fork in the subsequent version, and users will no longer be able to tell which is the main-net.

### 3.2.4 Mining Machine Monopoly

The POW consensus mechanism will inevitably lead to a race for mining machines. In order to obtain higher hash power, special-purpose mining machines with higher performance will be developed inevitably, and ordinary people cannot participate in mining. The CPoS mechanism is much more accessible because of slow iteration of hard disk manufacturers and low entry barrier. In traditional businesses, the vendor is normally not a competitor to users. But in the PoW systems, the ASIC manufacturer is

the biggest miner. It can be easily understood that the miner's competitor is the miner's vendor, since device suppliers take most of the profit by providing mining machines, miner is radically the risk-free arbitrage of ASIC manufacturers.

### 3.2.5 Power Resources Monopoly

The power resources monopoly leads to no PoW ecosystem expansion, as the cost of mining exceeds the return. For those CPoS miners, the hard disks have much lower power requirement, thus the return of mining is higher. The linear hedge ratio of civil computer hardware can also be taken into consideration to ensure that miners can hedge the price fluctuation risk in the secondary market under the condition of relative safety and cost protection.

## 3.3 BHD1 Architecture and Consensus Mechanism

The BHD1 wallet is derived from Bitcoin and the consensus from Burst's PoS2.

Bitcoin started in Jan 2009, the stability of wallet and blockchain is widely accepted after 10 years of iterations, it is safe and reliable to implement the PoS consensus on the Bitcoin QT wallet.

Burst Coin started in August 2014, and upgraded to PoS2 in 2018 after 4 years of iteration. Combining the advantages of Bitcoin and Burst, BHD1 has currently become the most reliable public chain with PoS consensus algorithm.

Since its launch on August 3rd 2018, BHD1 has grown steadily in computing power, withstood numerous tests, attacks, and cracks, and so far no major loopholes have emerged.

By adopting the mature PoS2 mechanism, a stable and reliable consensus mechanism is introduced to build community confidence in the BHD1 public chain. Since being compatible with Burst Plot files, miners can get both BHD1 and BurstCoin benefits, with only an additional operation.

The BHD1 wallet inherits Bitcoin's excellent P2P network architecture and UTXO system, which is mature and stable.

The wallet client could implement any latest developments from the Bitcoin community: lightning network, script upgrades, and much more. The interface standard is kept same as that of Bitcoin, allowing users to integrate easily.

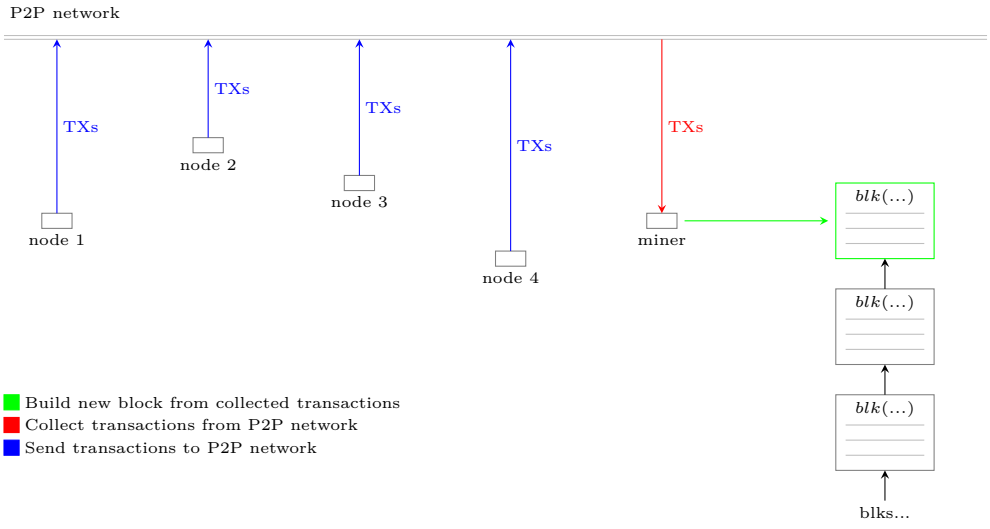
The CPoS ecosystem model includes mining pool, miner, crypto currency holder, wallet, exchanges and hardware vendor. The positive inner cycle and entrance of outside resources would bring expansion and development to this ecosystem, the rising price of BHD1 would attract more miners; more miners coming to the system will lead to further price increase.

The cost of PoW is influenced by four factors: cost of dishonesty, cost of mining, level of difficulty and cost of mining devices. In the end, the PoW would become another low gross margin industry, the former windfall profits was because of insufficient scale, fluctuation of secondary market and limited device vendors.

When it comes to PoS, due to the relatively low power consumption by hardware, miners can obtain other coins in the future symbiotic ecosystem of PoS almost free of charge without any risk.

The CPoS system, could give miners the choice to have most of the profits, incur cost for them to be the holder of other PoS coins, and avoid any malicious act. At the same time, the CPoS system attaches great importance to the release of distributional right and packaging right without barrier, which brings equity to the system. BHD1 network architecture and the participants:

**BHD1 network architecture**



### 3.3.1 Miners Mining Procedure

#### Plot

Miner plots file at local hard disk, and uses hash value to fill the disk. The larger the storage space, the more hash value could be filled, and higher block generation rate. Hash algorithm uses Shabal256, which is anti-ASIC.

#### Transaction

Wallet makes up the P2P network(inherited from BTC): Transactions happen between wallets.

#### Forging

Miner use wallet to listen to the P2P network, once a block is received, the packaging process of the next block starts. Wallet composes a block, sends the hash value of the block to miner, then miner finds the matching nonce. Once wallet receives nonce, it turns the nonce to deadline, wait for the time to end and then broadcast the block.

#### Verify

Receives the block, verifies it.

### 3.3.2 Plotting - Create Plot File

#### Algorithms and acronyms

**Shabal**: Shabal is the name of the crypto/hash function used in BHD1. It is a rather heavy and slow crypto compared to many other alternatives like SHA256. Thus Shabal is a good crypto for Proof of Space coins like BHD1, because we store the precomputed hashes while it is still fast enough to do smaller live verifications. BHD1 uses the 256bit version of Shabal, which is also known as Shabal256.

**Hash / Digest**: A hash or digest in this context is a 32Byte (256bit) long result of the Shabal256 Crypto.

**Nonce**: When generating a plot file, you generate something that is called nonces. Each nonce contains 256Kilobyte of data that can be used by miners to calculate Deadlines. Each nonce has its individual number. This number can range between 0-18446744073709551615. The number is also

used as a seed when creating the nonce, so each nonce has its own unique set of data. One plot file can contain many nonces.

**Scoop**: Each nonce is sorted into 4096 different places of data. These places are called scoop numbers. Each scoop contains 64byte of data which holds 2 hashes. Each of these hashes are xored with a final hash (we get to final hash while generating a nonce chapter).

**Plot ID**: When you create your plot file it will be bound to a specific BHD1 account. The numeric account ID is used when you create your nonces. Because of this all miners have different plot files even if they use the same nonce numbers.

### 3.3.3 Generating a Nonce

The first step in creating a nonce is to make the first seed. The seed is a 16byte long value containing the Plot ID and the nonce number. When this is done we start to feed the Shabal256 function to get our first hash.

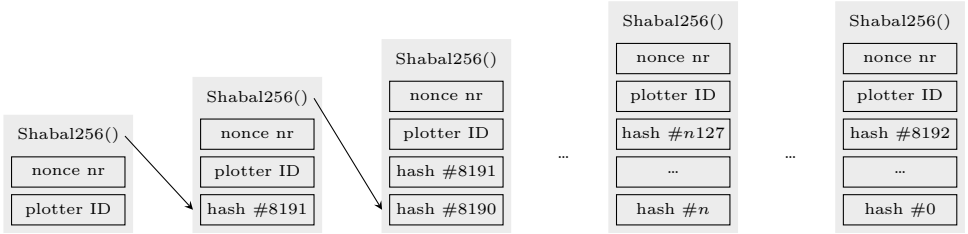
We have produced the first hash. This is the last hash in the nonce. Hash #8191. Now we take this produced hash (#8191) and pre-append it to the starting seed. The result will now be our new seed for the next round of shabal256 computation.

We now have produced two hashes.

Hash #8191 and Hash #8190. This time we pre-append Hash #8190 to the last seed we used. The result will now be a new seed to feed Shabal256.

Once again, we have created a new hash. This procedure of pre-appending resulting hashes to a new seed will continue for all 8192 hashes we create for a nonce. After iteration 128 we have reached more than 4096 bytes in the seed. For all remaining iterations we will only read the last 4096 generated bytes.

Once we have created 8192 hashes we are now going to make a Final hash. This is done by using all 8192 hashes and the first 16 bytes as seed.



The final hash will now be used to xor all other hashes individually.

We have now created our nonce and can store it in a plot file before we continue to the next nonce.

### 3.3.4 PoS Format

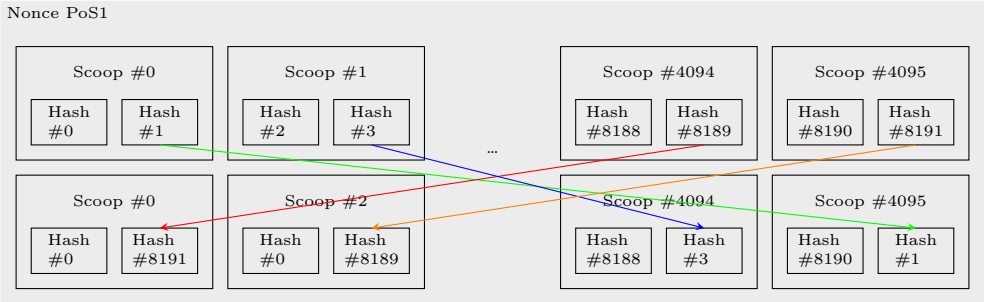
The PoS2 nonce format is created the same way as PoS1 with a slight addition to the end of the process. To create a PoS2 formatted nonce we need to shuffle the data around.

The data shuffling process:

Dividing the nonce in 2 halves, get a range with scoops 0-2047 and 2048-4095. Name 0-2047 the low scoop range and 2048-4095 the high scoop range. Take the second hash from a scoop in the low range, and swap it with the second hash in its mirror scoop found in the high range.

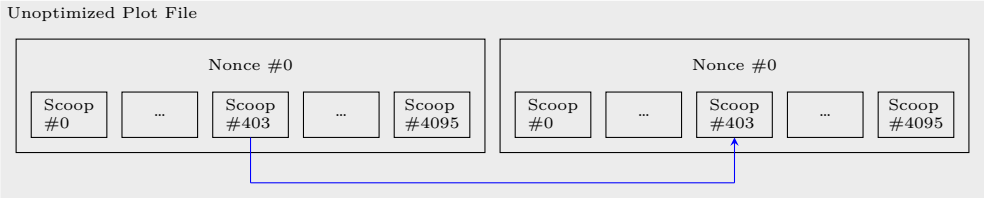
The mirror scoop is calculated like this:

$$MirrorScoop = 4095 - CurrentScoop \tag{3.1}$$

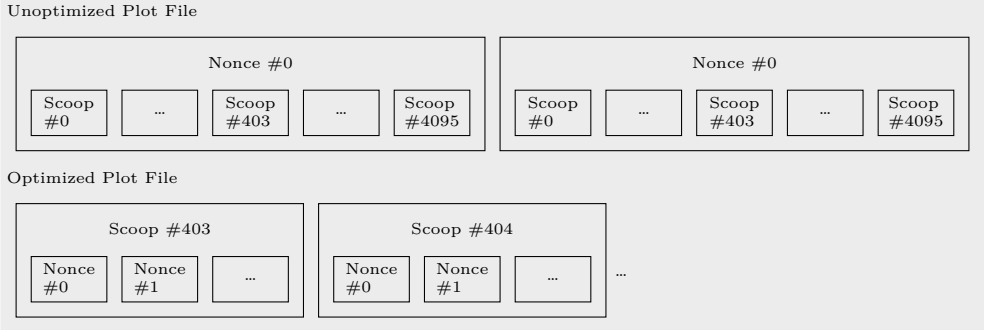



### 3.3.5 Plot Structure

When we are mining we read nonce from one or more plot files. The miner software will open a plot file and seek the scoop locations to read the scoops data. If the plot file is unoptimized, the scoop location will be on more than one place. In the following example the miner will be seeking and reading scoop #403.



This is not the most effective way since the miner will spend a lot of time to seek new locations on the storage device to be able to read the scoops. To prevent this, we can optimize plots or use plotter software that creates optimized plots from the beginning. Optimization is done by reordering the data in the plot file and grouping all data from the same scoop number together.




Basically, what we have done is to divide the plot file into 4096 portions where we split up all the nonces data based on scoop numbers. When the miner now wants to read Scoop 4096 it only seeks one time and read all data sequentially. This provides better performance.

### 3.3.6 Mining and Block Forging

#### Algorithms and acronyms

**Shabal / Sha256 / Curve25519**
Shabal, Sha256 or Curve25519 is the name of the crypto/hash function used in BHD1. Shabal is the main function in BHD1. It is a rather heavy and slow crypto compared to many other alternatives like SHA256. Thus Shabal is a good crypto for Proof of Space coins like BHD1 because we store the precomputed hashes while it is still fast enough to do smaller live verifications. BHD1 uses the 256bit version of Shabal, which is also known as Shabal256.

**Deadline**
When you mine and process Plot files, you end up with a value called deadline. These values represent the number of seconds that must pass since the forging of last block before block-forging is allowed. If no one else forges a block during this period, you can forge a block and get a block reward.

**Block Reward**

If you are lucky enough to cast a block, you will get BHD1. This is called a block reward. For every 420000 blocks, the block reward is reduced by 50%. The initial reward is 25 BHD1s per block, of which 1.25 belongs to the Foundation. Under full conditions, the miners'Union gets 23.75 BHD1s.

**Base Target**
Base target is calculated from the last 288 blocks. This value adjusts the difficulty for the miners. The lower the base target, the harder it is for a miner to find a low deadline. It gets adjusted in a way that BHD1 can have an average of 5 minutes for each block.

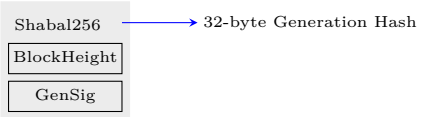
**Network Difficulty**
Network Difficulty, or NetDiff in short, is a value that can be read as an estimate on the total amount of space in Byte dedicated to mine BHD1. This value changes with every block in relation to base target.

**Block Height**
Every block forged gets an individual number. Every new block forged gets the previous block's number + 1. This number is called block height, and can be used to identify a specific block.

**Generation Signature**
Generation signature is a based from the previous block merkle root and block height, This value is then used by miners to forge a new block. Generation signature is 32 bytes long.

### 3.3.7 Mining Process

The first thing that happens when you start mining, is that the miner talks to the wallet and asks for mining information. This information contains a new generation signature, base target, and the next block height. Before the wallet sends over this info, it creates the generation signature by taking the previous generation signature together with plot id and runs this though shabal256 to get the new hash. The miner will now take the new 32 byte generation signature, and the 8byte block height, and put them together as a seed for Shabal256. The result will be a hash value called Generation hash.



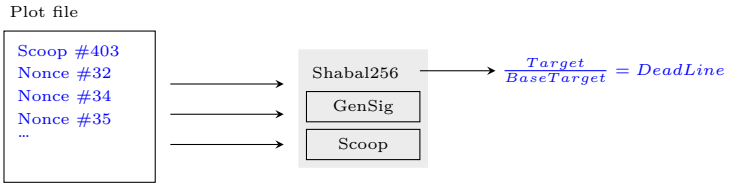
Now, the miner will do a small mathematical operation on this hash to find out which scoop number to use when processing the plot files. This is done by taking the generation hash modulo 4096, as there are only that many scoops

$$(GenHash)_{mod}(4096) = ScoopNumber \tag{3.2}$$

Next step for the miner is to read all the 64-byte long scoops from all nonces in all plot files. It will process them individually through shabal256 together with the new generation signature to get a new hash called target. This target is now divided with base target and the first 8bytes of the result is the value deadline.

$$Target = shabal256(scoopdata, generationsignature) \tag{3.3}$$

$$Deadline = \frac{target}{basetarget}; \tag{3.4}$$




To prevent so-called ”nonce spamming” to the wallet, the miner usually checks if the current deadline found is lower than the lowest one it has found so far. Usually there is also a max value that can be set, as ridiculously large deadlines are of no use to anyone. After these checks, the miner submits information to the wallet. This information contains the numeric plot ID bound to the plot file, and the nonce number that contains the scoop data used to generate the deadline.

### 3.3.8 Block Forging Process

#### Handling Deadlines

The wallet has now received the information submitted by the miner, and will now create the nonce to be able to find and verify the deadline for itself. After this is done, the wallet will now check and see if an equal amount or more seconds has passed as defined by the deadline. If not, the wallet will wait until it has. If a valid forged block from another wallet is announced on the network before the deadline has passed, the wallet will discard the mining info submitted since it is no longer valid. If the miner

submits new information, the wallet will create that nonce and check if the deadline value is lower than the previous value. If the new deadline is lower, the wallet will use that value instead. When the deadline is valid, the wallet will now start to forge a block.

**Forging**
The wallet will start by getting all of the unconfirmed transactions it has received from users or from the network. It will try to fit as many of these transactions possible until it hits the limit of 8M, or until all transactions are processed. For each transaction the wallet reads, it will do checks. For example, if the transaction has a valid signature, if it has a correct timestamp, etc. The wallet will also sum up all of the added transactions amounts and fees.

### BHD1 Wallet Vs BTC Wallet

BHD1 inherits the BTC wallet in code level, but differs from BTC in the following aspects:

1. BHD1 expands the block size to 2 MB/block, Blocks become larger, and a single block can contain more transactions to speed up transfers.

2. The avg block generation time is set to 3 mins. Block generation time is halved and the transaction time has increased.

3. Initial reward is set to 15 BHD1/block, halves each 4 years. The initial reward is halved, giving the community more time to develop, since early adopters are not taking too much advances, the miner community could share more profits.

| **Parameter** | **BTC** | **BHD1** |
|---|---|---|
| Total Supply | 21,000,000 | 21,000,000 |
| Block Time | 10mins | 3mins |
| Block Size | 1M | 2M |
| Halving Cycle | every 4 years | every 4 years |
| Initial Block Reward | 50 BTC | 15 BHD1 |

## 3.4 BHD1 Technical Characteristics

| |
|---|
| PoS2 consensus mechanism |
| 5-minute block generation time, the transaction speed is faster |
| 8 M block size to improve network efficiency |
| Zero knowledge proof will be added once whole network capacity reaches 3000P. Uses hard disk mining, anti-ASIC, mine without special equipment |
| Sustainable, low energy consumption, low noise |

### 3.4.1 Blockchain

Block includes proof sub block, signature sub block and transaction sub block. The arrow indicates that the sub-block contains the signature of the miner which has the arrow pointing to the sub-block. Our challenge is generated by sub-block hashed from blocks before current one.

### 3.4.2 Possible Attack and Prevention Design

While creating blocks, miners can try different transaction combinations, making the blocks created biased towards themselves. In our block structure, the independence of the proof sub-block prevents this attack.

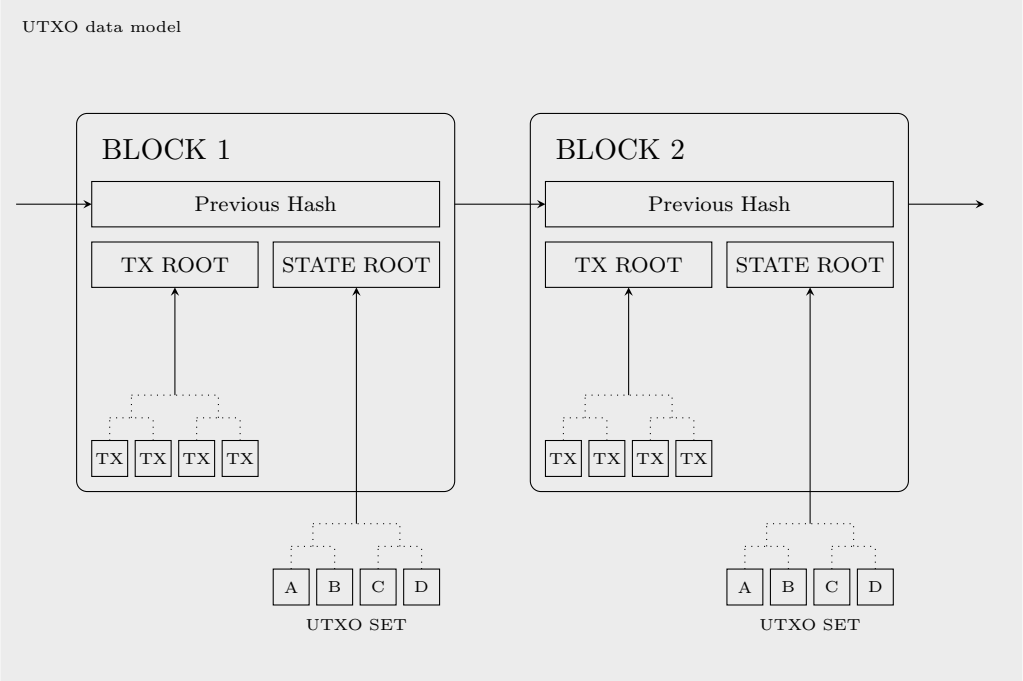
#### Challenge Grinding

- In the process of mining, miners can divide their space into m parts, then continuously refactor $t = 2\Delta$ on the blockchain.
- Then you can try $i_{th}$ block proof, making $i + \Delta$ quality the biggest. Based on linear summation calculated quality, according to the above attack method, it will result in an attacker having $\frac{1}{2}$ times chance to get bigger quality.
- By redefining the blockchain quality, the gains from this attack can be reduced. The calculation of quality is changed from linear superposition to multiplication.
- Under this definition, the probability increases obtained by the attacker will be reduced to $\log(m)$. At the same time, let the challenge of

continuous $\Delta$ block be determined by the same block, which will further reduce the impact of the attack.

### 3.4.3 Transaction

BHD1 transaction structure is the same as Bitcoin, that is, UTXO to UTXO's chain. This type of transaction design has also been available for many years, and it is also an effective way to achieve its basic properties.



chapterUpdate to Chia consensus

In the new version of BHD1, consensus mechanism is updated.

1. Consensus has been upgraded to Chia's PoST
2. Economic model is updated

## 3.5 Introduction

Decenterialized consensus algorithms require to consume scarce resources, such as computing power, staked money and storage space. BHD1 is continuing to use storage space to secure the network. Timelord has been added to provide a reliable cryptographic. By adding timelord to increase the security of entired network.

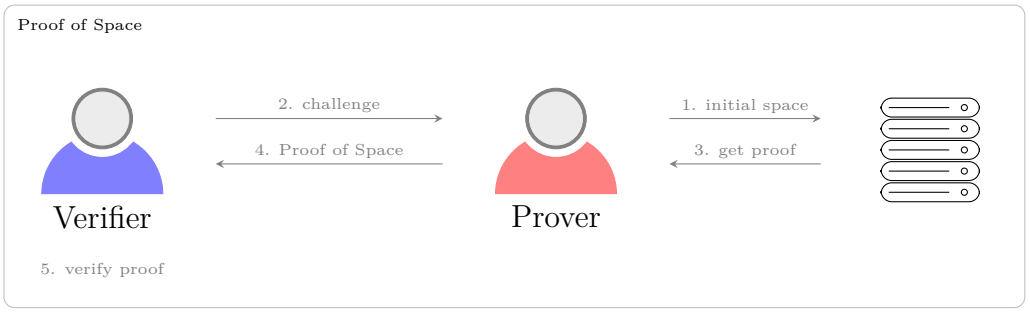
BHD1 cryptocurrency system combines Proof of Space (PoS) and Time (PoT). All required proofs are submitted to block-chain and can be easily verified. Miners need to find the correct answer by searching those random-looking data to win the lottery. No funds, special hardware, registration or permission is required to join except a hard driver and internet connection.

## 3.6 Proof of Space

A Proof of Space protocol is one in which:

- A Verifier can send a challenge to a Prover.
- The Prover can demonstrate to the Verifier that the Prover is reserving a specific amount of storage space at that precise time.

The Proof of Space protocol has three components: plotting, proving/farming, and verifying. For more info, see Chia's Details of the chiapos specification, and reference implementation by visiting https://chia.net.




### 3.6.1 Plotting

Plotting is the process by which a Prover, who we refer to as a miner, initializes a certain amount of space. To become a miner, one must have at least 101.4 GiB available to reserve on their computer (the minimum spec is a Raspberry Pi 4). There is no upper limit to the size of a Chia farm. Several farmers have multi-PiB farms.

As of Chia 1.2.7, a k32 plot can be created in around five minutes with a high-end machine with 400 GiB of RAM, or six hours with a normal commodity machine, or 12 hours with a slow machine using one CPU core and a few GB of RAM. Opportunities still remain for huge speedups. Furthermore, each plot only needs to be created once; a miner can farm with the same plots for many years.

Plot sizes are determined by a k parameter, where $space = 780 * k * 2^{k-10}$, with a minimum k of 32 (101.4 GiB). The Proof of Space construction is based on Beyond Hellman, but it is nested six times (thereby creating seven tables), and it contains other heuristics to make it practical.

Each of the seven tables in a plot is filled with random-looking data that cannot be compressed. Each table has $2^k$ entries. Each entry in table i contains two pointers to table i-1 (the previous table). Finally, each table-1 entry contains a pair of integers between 0 and $2^k$, called "x-values." A Proof of Space is a collection of 64 x-values that have a certain mathematical relationship. The actual on-disk structure and the algorithm required to generate it are quite complicated, but this is the general idea.

Once the Prover has initialized 101.4 GiB, they are ready to receive a challenge and create a proof. One attractive property of this scheme is that it is non-interactive: no registration or online connection is required to create a plot using the original plot format. Nothing hits the blockchain

until a reward is won, similar to PoW. For pool portable plots, a miner only needs a few mojos to create a plot NFT before plotting and then everything has the same characteristics from there.

### 3.6.2 Farming

Farming is the process by which a miner receives a sequence of 256-bit challenges to prove that they have legitimately put aside a defined amount of storage. In response to each challenge, the miner checks their plots, generates a proof, and submits any winning proofs to the network for verification.

For each eligible plot (explained later), a miner uses the following procedure to generate a full Proof of Space. Keep in mind that a plot consists of 7 tables (T1-T7) of approximately the same size, as well as 3 checkpoint tables (C1-C3), which are much smaller:

1. The miner receives a challenge from the block-chain
2. For each eligible plot, extract a k-sized value from the challenge, where k denotes the size of the plot (k32, k33, etc)
3. Look in the C2 table for a location at which to start scanning the C1 table
4. Scan the C1 table for the location at which to start scanning the C3 table
5. Read either one or two C3 parks. The number of parks to read depends on the index and value calculated from the C1 table. This requires an average of 5000 reads (the maximum is 10 000). These are sequential reads of 4 bytes (for an average total of 20 KiB)
6. Grab all the f7 entries matching the challenge value (which can be 0 or more), along with the index in the table at which they were found
7. For each matching f7 value, read T7 at the same index where the f7 value was found in its own table, and grab that entry, which is an index into T6
8. The T6 index contains one line point with two back pointers to T5, four to T4, eight to T3, sixteen to T2 and thirty-two to T1. Each back pointer requires 1 read, so a total of 64 disk reads (1 index from T7, 63 back pointers) are performed to fetch the whole tree of 64 x-values.

Since most proofs generated by this process are not good enough to be submitted to the network for verification, we can optimize this process by only checking one branch of the tree. This branch will return two of the 64 x-values. The position of the x-values will always be consecutive and will depend on current challenge (eg x0 and x1... or x34 and x35). We hash these x-values to produce a random 256-bit "quality string." This is combined with the difficulty and the plot size to generate the required_iterations. If the required_iterations is less than every required_iterations those are found from local storage or internet, the proof can be included in the blockchain. At this point, we look up the whole Proof of Space.

By only looking up one branch to determine the quality string, we can rule out most proofs. This optimization requires only around 7-9 disk seeks and reads, or about 70-90 ms on a slow hard drive.

**INFO**

*"Throughout this website, we'll make a simple assumption that a single disk seek requires 10ms. In reality, this is typically 5-10ms, so we're using a conservative estimate.*

*The 10ms estimate also takes into account the time required to transfer data after the seek. While storage industry specs typically assume that large files are being transferred, this does not hold true for BHD1 farming, where proof lookups only require a tiny amount of data to be transferred. Therefore, it's safe to assume the transfer is almost instant."*

BHD1 also uses a further optimization to disqualify a certain proportion of plots from eligibility for each challenge. This is referred to as the plot filter. The current requirement is that the hash of the plot ID, challenge starts with 9 zeros. This excludes 511 out of every 512 plots. The filter hurts everyone equally (except for replotting attackers), and is therefore fair.

The plot filter effectively reduces the amount of resources required for farming by 512x – each plot only requires a few disk reads every few minutes. A miner with 1 PiB of storage (10,000 plots) will only have an average of 20 plots that pass the filter for each challenge. Even if these plots all are stored on slow HDDs, and connected to a single Raspberry Pi, the average time required to respond to each challenge will be less than two seconds. This is well within the limits to avoid missing out on any challenges.

Each plot file has its own unique private key called a plot key. The plot ID is generated by hashing the plot public key, the miner public key, and either the pool public key (for OG plots) or the pool contract puzzle hash (for pooled plots). The requirements for signing a Proof of Space depend on the type of plots being used. See the Plot Public Keys page for details on the keys used for plot construction.

In practice, the plot key is a 2/2 BLS aggregate public key between a local key stored in the plot and a key stored by the miner software. For security and efficiency, a miner may run on one server using this key and signature scheme. This server can then be connected to one or more harvester machines that store the actual plots. Farming requires the miner key and the local key, but it does not require the pool key, since the pool's signature can be cached and reused for many blocks.

### 3.6.3 Verifying

After the miner has successfully created a Proof of Space, the proof can be verified by performing a few hashes and making comparisons between the x-values in the proof. Recall that the proof is a list of 64 x-values, where each x-value is k bits long. For a k32 this is 256 bytes (2048 bits), and is therefore very compact. Verification is very fast, but not quite fast enough to be verified in Solidity on Ethereum (something that would enable trustless transfers between chains), since this verification requires blake3 and chacha8 operations.

## 3.7 Proof of time (VDFs)

A Verifiable Delay Function, also referred to as a Proof of Time or VDF, is a proof that a sequential function was executed a certain number of times.

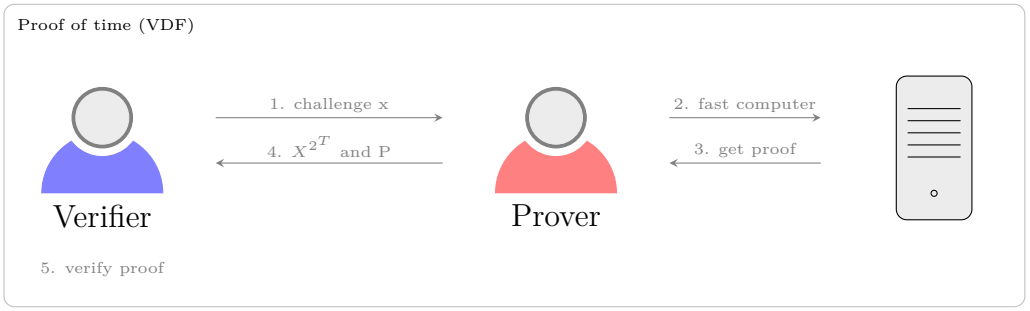
**Verifiable**: This means that after performing the computation (which takes time), the Prover can create a very small proof in a very short time, and the Verifier can verify this proof without having to redo the whole computation.

**Delay**: This means that the Prover actually spent a real amount of time (although we don't know exactly how much) to compute the function.

**Function**: This means it's deterministic: computing a VDF on an input x always yields the same result y.

The key word here is ”sequential”, like hashing a number many times: hash(hash(hash(a))), etc. This means the prover cannot just add more machines to make the function execute faster. Therefore we can assume that computing a VDF requires real (wall-clock) time. The construction that we use is repeated squaring. The Prover must square a challenge x T times. This requires time  (T). The Prover also must create a proof that this was performed properly.

Although the following details are not very important for understanding the consensus algorithm, the choice of what VDF to use is relevant, because if an attacker succeeds in obtaining a much faster machine, some attacks become possible.




The VDF used by BHD1 is repeated squaring in a class group of unknown order. There are two main ways to generate a large group that has an unknown order:

1. Use an RSA modulus, and use the integers mod N as a group. The order of the group is unknown if you can generate your modulus with many participating parties using an MPC ceremony.

2. An easier approach is to use classgroups with a large prime discriminant, which are groups of unknown order. This does not require any complex or trusted setup, so we chose this option for BHD1.

To create one of these groups, one just needs a large, random, prime number. The drawbacks are that classgroup code is less tested in real life, and optimizations are less well-known than in RSA groups. We use the same initial element for the squaring (a=2, b=1 classgroup element), and instead use the challenge to generate a new random prime number for each VDF, which is used as the discriminant. The discriminant has a size of 1024 bits, which means the proof sizes are around 1024 bits. We use the Wesolowski scheme split into n (1<=n<=64) phases so that creating the

proofs is very fast. Since the n-wesolowski proofs can be large, we replace them with 1-wesolowski proofs as soon as they are available. These are smaller, but require more time to make. The proofs themselves are not committed to on-chain, so they are replaceable.

### 3.7.1 Infusion

As a recap, VDFs take in an input, called a challenge, and produce an output, together with a proof that certifies that the function was evaluated correctly.

A value, in this context, can be thought of as a block with a Proof of Space. The value is combined with an output of a VDF, to generate a new value, which is used as the input/challenge for the next VDF. This is known as an infusion of a value into a VDF.

Therefore, we are chaining VDFs, but committing to a new value in between. This is used so that we have a linear progression of blocks, alternating proofs of space with proofs of time.

## 3.8 Blockchain

After BHD1 blockchain hard-fork, some of the properties have been changed according the consensus algorithms.

| Parameter | BTC | BHD1 | BHD12 |
|---|---|---|---|
| **Total Supply** | 21,000,000 | 21,000,000 | 63,000,000 |
| **Block Time** | 10mins | 3mins | 3mins |
| **Block Size** | 1M | 2M | 2M |
| **Halving Cycle** | every 4 years | every 4 years | every 4 years |
| **Initial Block Reward** | 50 BTC | 15 BHD1 | 30 BHD1 |

### 3.8.1 Challenge

The BHD1 consensus algorithm relies on timelords running VDFs for each block, which are adjusted periodically (and automatically) to take around 3

minutes. During every block, challenges are generated according previous block, and a sort of mini lottery starts, where farmers check their plots for proofs of space. When farmers find a Proof of Space that qualifies, they broadcast it to the network after the VDF calculation with required_iterations is finished.

A challenge is always a 256-bit hash. It is released when a new block has been added to blockchain. The challenge services both PoS and VDF.

### 3.8.2 Quality and iterations

#### Quality

The number of quality is used to represent the quality of a proof of space which is found from plots. According the quality, an iterations will be calculated and it controls how many times should be passed before miner can post the block with the PoS.

$$Quality = \frac{QualityString * Plot_{size}}{2^{256}} \tag{3.5}$$

#### Iterations

"required_iterations" is the number that VDF should run with. To verify a VDF proof not just only verify the proof itself, we also need to ensure the number of iterations is exceeded the requirement from PoS. Smaller of the number means the better quality it is. The block contains with the proof with better quality will be released earlier than others.

$$Iters = Difficulty * Difficulty_{factor} * \frac{QualityString}{2^{256} * Plot_{size}} \tag{3.6}$$

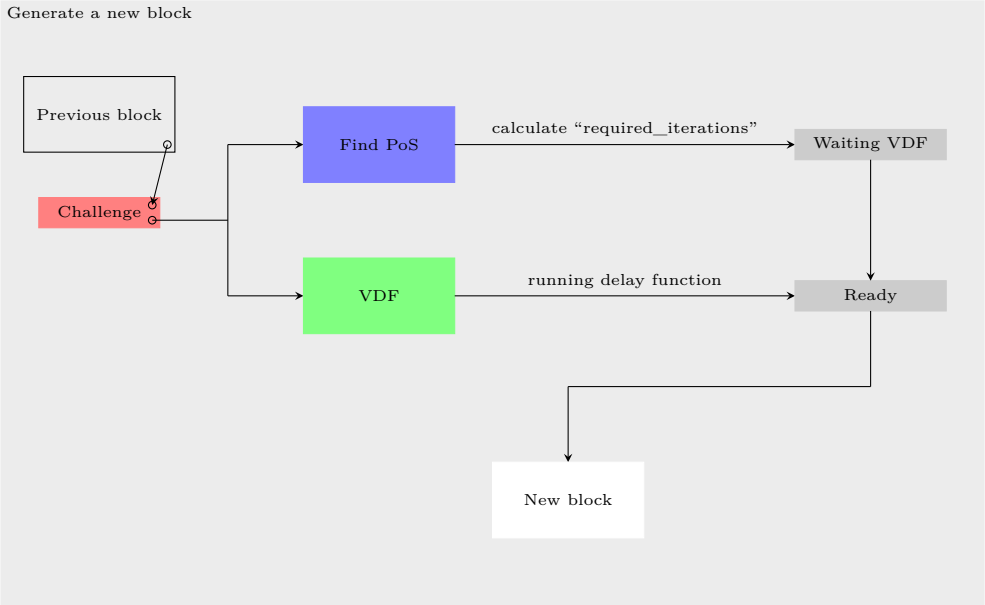
QualityString is mixed from current challenge and proof. $\frac{QualityString}{2^{256}}$ is a number between 0 and 1, multiply the number with the size of plot file will get the quality of the proof. The quality is used to multiply with difficulty in order to get the number of iterations. $Difficulty_{factor}$ is a constant number to fix the iterations to a reasonable range.

### 3.8.3 Blocks

BHD1 releases block every 3 minutes. The basic information are assembled into the block such as transactions, Proof of Space, VDF proofs and difficulty.

### Block generation steps

1. Find Proof of Space from plots
2. Calculate “required_iterations” according the proof
3. Wait and retrieve VDF proof from timelord
4. Create new block and pack with all proofs and TXs etc




### Void block

There is a very rare situation, the proof of space cannot be found from the entired network. The consensus will add an empty duration without a PoS called void block. After the proof of VDF is calculated, miner will be asked to mix a new challenge to find a Proof of Space. The void block will be included into the new block.

## 3.8.4 Difficulty

Difficulty is the value represents how good is the block. According to current netspace and VDF speed, difficulty will be adjusted on every block. The time to release a new block will be around 3 minutes.

#### Difficulty adjustment

The equation of the difficulty adjustment is trying to calculate the new weight of next block. Adjusting difficulty also affects the number of iterations, this is also the way how the new block releases after 3 minutes.

$$NewDifficulty = \frac{Weight_{current} - Weight_{previous}}{Time_{CurrentBlock} * Time_{PreferBlock}} \tag{3.7}$$

### 3.8.5 Network space

Network space is the value that represents total amount of space those are allocated to generate current blockchain. Network space can be calculated from the difficulty of the last block on the chain.

$$NetSpace = \frac{Difficulty_{current}}{Iters_{current}} * DifficultyConstantFactor * 2^{FilterBits} \tag{3.8}$$

*Please note: the calculated size is not exactly the value of the network space. It is just the rule to mesure the network.*

## 3.9 Block validation

### 3.9.1 Block format

#### Proof of Space

To verify a PoS, the "PlotId" from plot is required, and we also need the public-key of the farmer (aka "farmer-pk") to verify the signature later. The plot also needs to be verified to ensure it is owned by the farmer. Record all related fields is the best way to accomplish it.

| Name | Data type | Description |
|---|---|---|
| Pool pk or Hash | 48 or 32 bytes | According Chia's consensus. |
| Pk type | 1-byte | The type of the public key (OG-Plot or PooledPlots) |
| Local pk | 48-byte | Local public-key identify the plot provide the proof |
| Farmer pk | 48-byte | Identify the farmer |
| K | 1-byte | The size of the plot file |
| Proof | multi-bytes | The proof of space |

### VDF proofs

Verify VDF proof is more easier. Provide "Proof", "Y", "witness type" and "Iterations" to verify function will get the result. And the verifier also need to ensure the number of iterations is enough to satisify the consensus.

| Name | Data type | Description |
|---|---|---|
| Y | multi-bytes | large prime discriminant |
| Proof | multi-bytes | The proof of time |
| Witness type | 1-byte | The type of witness |
| Iterations | 64-bit | The number of iterations |

### Farmer signature

Farmer signature is used to ensure the owner of the proof of space. The signature can be verified by "farmer-pk". The number of bytes of the signature is 96.

### Quality

A 64-bit number represents the quality of proof of space. The way to verify the quality is use the equation we mention before.

**Difficulty**

Difficulty is a 64-bit constant number represents the network space. The difficulty can be calculated from previous block.

### 3.9.2 Verify block

The following steps list all required checks to ensure the validity of a block.

1. Check previous block - To ensure the challenge is correct and generated from the previous block
2. Check duration of VDF - The duration between blocks must has a reasonable value
3. Check number of iterations of VDF - The number of iterations must satisify the requirement
4. Check void blocks - Ensure the void blocks are valid and the challenge is mixed correctly
5. Check difficulty - The difficulty can be calculated from the previous block
6. Check the quality of the proof
7. Verify the proof of space
8. Verify the proof of VDF
9. Verify farmer's signature
10. Check distributed amount by coinbase
11. Check validity of pledges TXs

## 3.10 Economic model

Economic has been improved after the Chia's consensus is updated. The total supply amount is increased and add "lock period" for each pledge.

### 3.10.1 Total supply is increased

The amount of total supply is increased to 63,000,000.

- The total amount to be supplied in each block will be doubled after consensus updated
- The foundation will receive an additional amount of the total amount multiply with 2 already supplied on the current chain at one time

### 3.10.2 No more fund to foundation

The Foundation will no longer receive funds in future blocks.

### 3.10.3 Pledge improvement

The miner needs to pledge a certain amount of BHD1 to the chain, and the pledged amount is related to its pledge time. When the pledge time is less than three years, the pledge amount will be discounted. During the pledge time, these BHD1 will not be able to be withdrawn.

#### Netspace

Assuming that the netspace of a miner is "p", the current netspace of the entire network is "t", and the current distributed amount is "m", then the current miner wants to achieve the condition of full pledge, and the amount of currency "a" needs to pledge is:

$$a = \frac{p}{t} * m \tag{3.9}$$

#### Lock period

Now the miner needs to select the type of period for every new pledge. It determines the ratio value of the total amount that the pledge amount actual is.

| **Period** | **Ratio** | $1_{BHD1}$ **required** |
|---|---|---|
| Current Deposit (1 week) | 8% | $12.5_{BHD1}$ |
| 1 year | 20% | $5_{BHD1}$ |
| 2 years | 50% | $2_{BHD1}$ |
| 3 years | 100% | $1_{BHD1}$ |

### Burn

When the pledge period has not yet expired, it is allowed to lose part of the pledged currency to withdraw the pledged amount, but part of the currency will be destroyed according to the pledged time. Assuming that "p" blocks have been pledged, a total of "f" blocks need to be pledged, and the amount of pledged currency is "a". Then, withdrawing the pledge with the amount of currency a in block "p" will return the amount "w" back, and $a - w$ is the amount of currency destroyed after this withdrawal.

$$w = \frac{p}{f} * a \tag{3.10}$$

### Burning mechanism

BitcoinHD1 will use a special (20-byte) accountID, it is filled by 0x23 of each byte. No one owns the private-key, and the consensus also prevent that no one will be able to make a new transaction that transfer BHD1 from this account even the foundation. All burned BHD1 will be sent to this account, and the amount required of mining will be re-calculated on next block.

### Total amount update frequency

The number of total distributed amount is calculated once a month (total 20*24*30 heights), and the data will remain unchanged for one month until the next calculation. This means that the amount of currency that all miners need to pledge also changes every month. This mechanism simplifies the calculation of pledges, and miners no longer need to calculate the total distributed amount frequently.

### Insufficient pledge amount

When the amount of pledge is not enough for a miner to claim all rewards from a new block, miners will only get 15% of the block rewards, the rest 85% will be locked in the chain until the miner who has enough amount of pledge to claim all rewards.

### For example

We assume that there is a miner initialized $15_{PB}$ storage to do the mining. Assume that the current netspace of the entire network is $200_{PB}$ and the total supplied amount is $10,000_{BHD1}$. Thus, according to the formula, the total amount that needs to be pledged per PB is $\frac{10,000}{200} = 50_{BHD1}$. The

miner in order to obtain 100% mining rewards, a total of $50_{BHD1} * 15_{PB} = 750_{BHD1}$ needs to be pledged to the chain.

100% **rewards**

The miner can choose one of the following pledge plans to get 100% rewards:

1. Deposit $9375_{BHD1}$ to the chain with period "Current Deposit"
2. Deposit $3750_{BHD1}$ to the chain with period "1 year"
3. Deposit $1500_{BHD1}$ to the chain with period "2 years"
4. Deposit $750_{BHD1}$ to the chain with period "3 years"

15% **rewards**

Miner will receive 15% rewards when the amount of pledge is less than $750_{BHD1}$. The pledge amount also increases when the total supply is increased or the miner initialized more space to do the mining.

**Withdraw an unexpired pledge**

The withdrawal amount will be calculated according to the percentage of the pledged period when the pledge is withdrawn unexpired. The remaining amount will be directly burned on the blockchain. For example: there is a pledge with total amount of $300_{BHD1}$, the period of this pledge is 3 years. If we want to withdraw it after only 10 months. There are only a total of $83.33_{BHD1}$ will be withdrawn, the remaining $216.67_{BHD1}$ Will be burned.

Formula to calculate the amount

$$Withdrawal_{BHD1} = Total_{BHD1} * \frac{Time_{Elapsed}}{Time_{Agreed}} \tag{3.11}$$

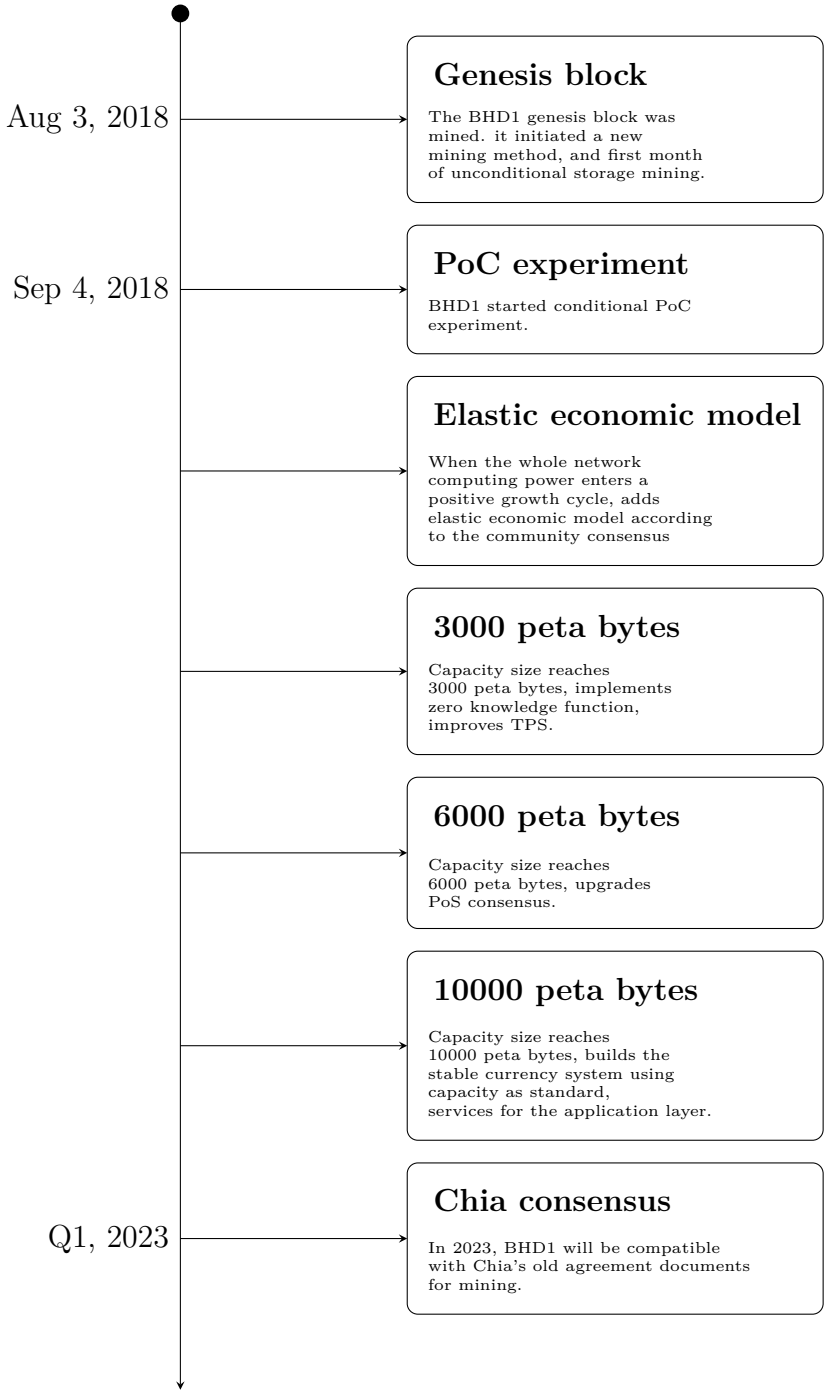
### 3.10.4 Foundation addresses

There is at least one foundation address to be able to use for generating new blocks without binding them with a valid farmer public-key. This is a mechanism to ensure that the network will keep generating new blocks and this allows new miner to create binding/pointing transaction to add miners to the network. Consensus ensures that the pledge amount of foundation addresses will not be able to calculate with full mortgage, even someone deposit pledge to the foundation addresses.

# Chapter 4

# Tech Roadmap

1. August 3, 2018: The BHD1 genesis block was mined. It initiated a new mining method, and first month of unconditional storage mining.
2. Sep 4, 2018: BHD1 started conditional PoS experiment.
3. When the whole network computing power enters a positive growth cycle, adds elastic economic. Model according to the community consensus.
4. Capacity size reaches 3000 peta bytes, implements zero knowledge function, improves TPS.
5. Capacity size reaches 6000 peta bytes, upgrades PoS consensus.
6. Capacity size reaches 10000 peta bytes, builds the stable currency system using capacity as standard, services for the application layer.
7. In 2023, BHD1 will be compatible with Chia's old agreement documents for mining.




*BHD1 is committed to becoming a high value financial system that changes the way crypto currencies are produced.*