Haven Protocol
Private Decentralized Finance

Core Protocol v3.0

This paper is intended to document the core functionality offered by Haven Protocol. Other second layer functions are not covered in this paper, and will be addressed separately on a case by case basis.

Introduction

Bitcoin paved the way for electronic peer-to-peer currency. It was the first digital currency to successfully implement a distributed ledger of transactions based on cryptographic proof over trust. More recently, with the realisation that all wallets and transactions in many cryptocurrencies are visible to all who care to look, the demand for private transactions and privacy coins has grown. Haven is built on top of Monero, which is widely considered to be the leader in privacy technology. Haven therefore inherits all of Monero's privacy features, including ring signatures and Bulletproofs. It extends that functionality by providing private, anonymous, synthetic currencies and commodities (xAssets) which can only exist through the "burning" of the Haven base currency – XHV. Haven also extends Monero's proof of fungibility, to allow for multiple asset types to be equated based on their monetary value rather than only the number of coins exchanged, creating the first of its kind, fully private set of synthetic currencies and assets.

Welcome to Haven - Private Decentralized Finance.

Project History

The concept of Haven was started by two developers in early 2018. This first attempt reached the stage of a public testnet before weaknesses in the solution, a hiatus in development, and a subsequent lack of progress from the original developers put the project's future in doubt. In late January 2019, a collection of original Haven community members took the project over with a view to completing the project, delivering the offshore storage mechanism, and building out the supporting infrastructure to gain mass adoption of a much needed utility in the rapidly growing cryptocurrency market.
The mainnet of Haven Protocol launched successfully on July 20th 2020 introducing it's first private currency xUSD to the market.

Haven Protocol

The promise: 1 xUSD will always be redeemable for $1.00 worth of XHV.

i. Concept

Haven is an untraceable cryptocurrency with a mix of standard market pricing and real world asset-pegged value storage. It achieves this via a ‘mint and burn’ process within a single blockchain. In the simplest case, users can burn Haven (XHV) for the equivalent USD value worth of Haven Dollars (xUSD). Or, to restore to a volatile state, the user can equally burn xUSD for $1 USD worth of XHV.

Other major fiat currencies including GBP, EUR and CNY, as well as Silver, Gold and other high profile commodities such as Oil are intended to be added to the Haven ecosystem over time to allow users to choose a suitable pegging mechanism for their needs.

ii. The Offshore Process – “Mint and Burn”

Haven uses a system called “mint and burn” to maintain its value relationship against its asset pegs. In practice, using the synthetic U.S. Dollar (xUSD) as an example, this works as follows: Bob decides he wants to put 200 of his Haven (XHV) into Offshore Storage. When users put XHV into Offshore Storage, they are burning XHV coins and minting the current value of those XHV as new xUSD. Offshore Storage determines the current market value of that Haven (in xUSD) based on a weighted average of volume across supported exchanges. This is done using a pricing oracle (a mechanism to discover real world data and make this data available to a blockchain) to retrieve pricing data for the full Haven ecosystem and create pricing records.

If the current value of Haven is $1 USD, offshore storage will burn Bob’s 200 XHV by constructing a special transaction where the 200 XHV that was sent is then burned into xUSD and the total supply of XHV decreases. If the market price of XHV then moves to $2 USD and Bob decides to access his offshore storage, he will be returned 100 XHV (100 * $2 = $200 USD as per original value).

If the opposite occurs and the price of Haven halves to $0.50, then 400 XHV will be minted and sent to Bob (400 * $0.50 = $200 USD as per original value). Clearly, the use of mint and burn therefore alters circulating supply of the underlying assets in a dynamic manner.

This creates intriguing supply scenarios – very different from other cryptocurrencies – which need to be reviewed by readers thoroughly in order to fully understand the Haven Protocol concept.

iii. How Does Offshoring Actually Work?

The Haven Protocol enables offshore transactions within the Haven Vault using a ‘coloured coin’ model. It is the first working implementation of coloured coins on the Cryptonote protocol. The concept of coloured coins is well known and defined within the Bitcoin network, and is described as far back as 2013 here:

https://www.coindesk.com/colored-coins-paint-sophisticated-future-for-bitcoin

Coloured coins on cryptonote however cannot work in the same way as Bitcoin, and in fact the concept of coloured coins within Cryptonote must be reworked and reimagined. Thanks to Nate Eldredge for this clear description of the differences between implementing using Bitcoin and Monero:

> *“With Bitcoin, there is a one-to-one correspondence between inputs and outputs of transactions. Suppose there is a transaction X with an output X1 that sends 1 satoshi to Alice's address A, and everyone agrees that output X1 is colored so that it grants title to Alice's 1977 Chevy Nova. If Alice decides to give the car to Bob, she creates a new transaction Y, with an input pointing to X1, and whose sole output Y1 sends 1 satoshi to Bob's address B. Now Bob can prove, by creating a signature corresponding to his address B, that he is the rightful owner of the car.*
>
> *If Mallory tries to claim the car by creating a different transaction with input X1, she will be found out, because she can't sign that transaction with Alice's private key, so it won't verify. If Alice tries to give the car to someone else by creating a second properly signed transaction Z with input X1, it will be detected as a double spend because another transaction spending X1 precedes it in the blockchain.*
>
> *With ring signatures, this correspondence is broken. When creating a transaction, in addition to the one output (of a previous transaction) that you really want to spend, you can list many others. You create a signature that proves that you are authorized to spend one of the outputs you listed, but doesn't give any information about which one it was. However, a linking algorithm ensures that any future attempt to spend that output again will be noticed and rejected.*
>
> *In the above scenario, if Alice uses a ring signature on her transaction Y, including not only X1 but another output Z1, then her signature will not prove that she is entitled to spend X1 (and therefore is the rightful owner of the car and can give it away); it only proves that she is entitled to either X1 or Z1.*
>
> *Furthermore, Mallory could create a transaction M that includes X1 and another output K1 that she is entitled to spend. Since she has the private key corresponding to K1, she can properly sign the transaction M, but it won't be clear whether it is spending X1 (which would convey title to the car) or K1 (which won't).”*

The above description describes the way coloured coins have been viewed and implemented within the Bitcoin network, and rightly points out that this model fails when both X1 and Z1 are still in existence after the initial transaction. Haven however, works slightly differently. Haven has no Alice, and we also have no Mallory. All we have is Bob.

When Bob converts from XHV to xUSD he sends a transaction with two colours, X (XHV) and Z (xUSD). The transaction takes as inputs coins of only the first color X, and has outputs of both X and the second colour Z. Each transaction within the Haven network contains two values for each destination (#X,#Z), and for all transactions, only one of these values can be non-zero for each destination.

So when Bob converts his 200 XHV at a price of $1.00 per XHV he sends a transaction with inputs of (200,0) and destination values of (0,200) giving an output of 200 xUSD and 0 XHV. If the price of XHV then moves to $2 per XHV then the conversion back to XHV would send a transaction with inputs of (0,200) and destination values of (100,0) giving an output of 100 XHV and 0 xUSD. In this way, inputs to transactions and UTXOs are permanently and effectively burnt atomically and in real time during the transaction process, and outputs are minted similarly.

This is all great, however Haven is a fork of Monero and inherits all of it’s security and anonymity features… and Monero is built on the premise, condition and absolute surety that for any given transaction; the difference between inputs and outputs is zero. Any transaction which does not satisfy this requirement will always fail.

In the case of Haven, this fundamental aspect of Monero cannot be true, and in fact for any and every exchange between XHV and xUSD where the price of XHV is not precisely $1.00 this rule is completely broken, inputs and outputs will not be equal, neither will our commitment sums of C^a and C^b and consequently *src/ringct/ rctSigs.cpp verRctSemanticsSimple()* will fail the Monero test for:

$$\sum_j C_j^{'a} - \sum_t C_t^b = 0$$

Here we introduce the concept within the Haven network of 'Proof of Value'.

Thanks are given to the Monero Research Lab for their paper on Concise Linkable Ring Signatures and Forgery Against Adversarial Keys [Brandon Goodell, Sarang Noether and Arthur Blue] https://eprint.iacr.org/2019/654.pdf ['the paper'] which has been used as part of the Haven implementation of Proof of Value.

In an early draft of 'the paper', the writers proposed a 'toy' model where they create a coloured currency with a fixed peg between two colors: dollars and pennies with a 100 : 1 exchange rate between them, and show how this can be done using CLSAG. The process is as follows:

1. Define an exchange rate by determining a constant ξ and some constants γC , γD on 1, 2, . . . , 2 ξ−1, (in this example, γC = 100 and γD = 1).
2. Alter the commitment structure so that each commitment is now a pair of commitments C and D for their corresponding colours
3. Create a range proof from Prove covering the values of both C and D. Here, C and D play the role of the Zj points, and P is additional data required for the transaction protocol.
4. We say a simple transaction key is valid if the following are satisfied:
    a. every input ring member (Xi , Ci , Di , Pi) ∈ Q has a valid range proof Pi so Ver(Pi) = 1; and
    b. every output range proof P 0 k is valid so Ver(P 0 k) = 1; and
    c. for the modified ring pk = X1 X2 · · · Xn Z1 Z2 · · · Zn the signature σ passes the 2-CLSAG verification, Verify(m, pk, σ) = 1 .

The effect of this is that the transaction is signed not with a commitment to zero, but a commitment to a difference - that difference being the difference in number of 'coins/tokens' this transaction creates based on inputs and outputs. If a user was exchanging 1 USD for 100 pennies, the difference would be 99 - the number of new coins minted. This model works because in order for a sender to sign using the difference that user MUST know both the number of coins used as inputs (which only the holder of the private key to those inputs can know) and they must use the correct exchange rate of 100:1, with all bulletproofs holding both possible colours' values. By doing so, they can correctly sign using the difference between inputs and outputs, and the transaction will validate.

The above model has one major flaw when considering the Haven mint and burn system. It requires a fixed exchange rate. Fixed and known by both sides of the transaction, and also fixed and known by any and all validators of the transaction. This creates a problem for us, and this model will not work because by definition to peg a volatile asset to a stable one, the thing that must change is the exchange rate.

iv. Proof of Value.

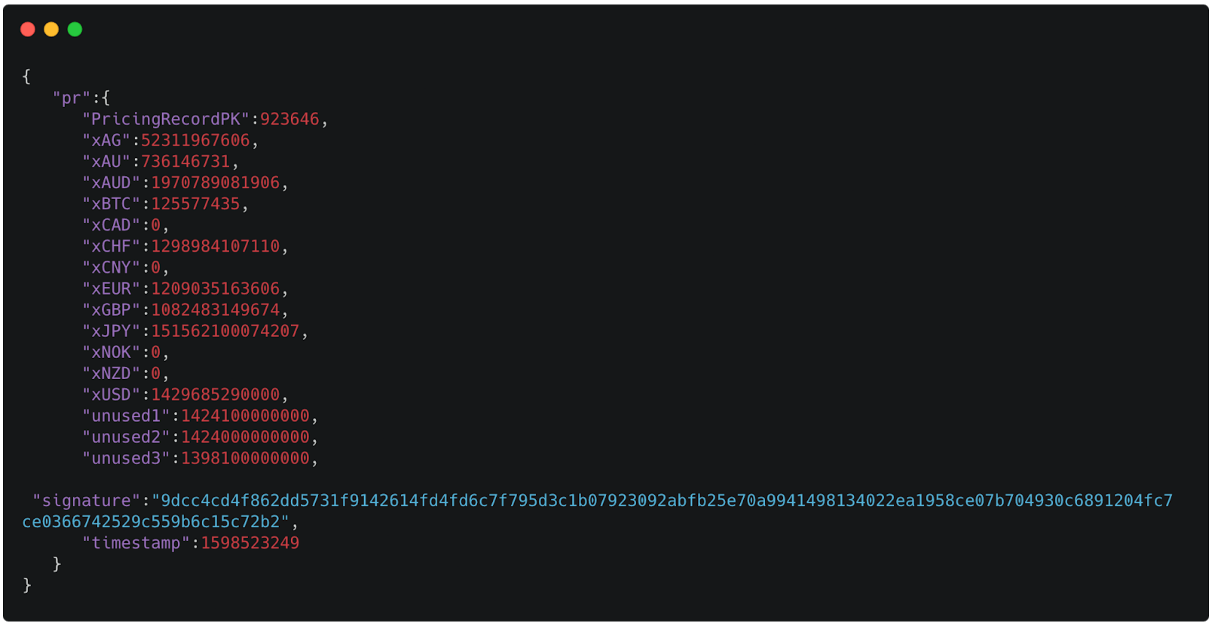
To make the above model of coloured coins work with a variable exchange rate requires:

1. A way to gather agreed and immutable pricing information, so that at any given time, an exchange transaction can use a price which can be validated
2. A way to convert inputs into outputs based on that price
3. A way to validate that the sender of a transaction satisfies the same requirements as any other cryptonote transaction - namely that they know the secret key to the inputs used, and can therefore convert using an exchange rate and sign a transaction with a correct difference
4. A way to validate that the agreed price has indeed been applied to the exchange, without disclosing any amounts to validators.

Pricing details are obtained from a real-world pricing provider (i.e. a pricing oracle) and a pricing record is created in preparation for a new block being solved. Pricing records contain the exchange rates (against XHV) for each of the xAsset pegs at the time of the block being mined. The pricing information is updated at 30 second intervals, and presented to the Haven daemon upon request. Pricing records are embedded into the blockchain in every block header by the miner solving that particular block.

By including this information in every block, the protocol guarantees that the transaction value cannot be tampered with or altered in any way – the blockchain guarantees that the pricing information is immutable. If multiple blocks are successfully mined within the 30 second lifetime of the current pricing record, the same record will get included in multiple blocks.

A pricing record contains the following conversion rates (all against XHV), as well as some reserved space for future additions and the signature of the oracle providing the data. An exemplar pricing record is:



```
{
   "pr":{
      "PricingRecordPK":923646,
      "xAG":52311967606,
      "xAU":736146731,
      "xAUD":1970789081906,
      "xBTC":125577435,
      "xCAD":0,
      "xCHF":1298984107110,
      "xCNY":0,
      "xEUR":1209035163606,
      "xGBP":1082483149674,
      "xJPY":151562100074207,
      "xNOK":0,
      "xNZD":0,
      "xUSD":1429685290000,
      "unused1":1424100000000,
      "unused2":1424000000000,
      "unused3":1398100000000,

 "signature":"9dcc4cd4f862dd5731f9142614fd4fd6c7f795d3c1b07923092abfb25e70a9941498134022ea1958ce07b704930c6891204fc7
ce0366742529c559b6c15c72b2",
      "timestamp":1598523249
   }
}
```

Pricing Record Example: Carbon

2/ Haven does this in the example above [Bob] using commitment pairs rather than single commitment values. This is also the method used in the toy example from Monero research labs.

3/ Haven transactions are signed using CLSAG and paired bulletproofs as described above. However, we do not sign using the difference as in the toy example. We sign using the original commitment to zero values. Our commitment is to a zero difference in **value.**

4/ This is where it gets complicated. To understand how Haven validates or rejects transactions using a proof of value requires a little bit of pre-work and some understanding of Public Key algorithms, and of how Cryptonote uses Elliptic Curve operations and points to validate input and output amounts.

Every transaction passes through the function *verRctSemanticsSimple()* which sums all the inputs and outputs of a transaction to check that the results are equal. Although the values are at this stage fully encrypted and represented as Elliptic Curve [‘EC’] points rather than real numbers, these sums still work due to the properties of modular arithmetic and the specific way Monero’s EC points are chosen/generated.

In short, although the numbers are encrypted, they still hold certain properties - the differences between them (within EC space) are still valid, and so a zero difference will still be a zero difference because commitments are additive.

In other words, if we had a transaction with inputs containing amounts $a_1, ..., a_j$ and outputs with amounts $b_1, ..., b_k$, then an observer would justifiably expect that:

$$\sum_j a_j - \sum_k b_k = 0$$

For Haven this would still work for XHV transfers, and xUSD transfers, but for exchanges this is completely incorrect.

So, re-using some notation from above, lets define constants γC , γD as the exchange rate for a *single* transaction, that exchange rate being provided by our pricing oracle. And now with commitments paired in our range proofs being (C, D) respectively. To prove equality of value we require that the sum of the value of inputs equals the sum of the value of outputs.

Our validation now looks like:

$$\lambda C \left(\sum_i C_i - f_c G - \sum_k C_k^{'} \right) = 1/\lambda D \left(\sum_i D_i - f_D G' - \sum_k D_k^{'} \right)$$

Where λC, λD signify that the bracketed values are summed based on their respective exchange rates. (C,D) signify input commitments, (C’,D’) signify output commitments, and fxG signifies the fees paid.

v. Pricing Oracles

In order to retrieve data from the real world, blockchains use a construct called an “oracle.” “A blockchain oracle is a third-party information source that has the sole function of supplying data to blockchains”

Source: *https://www.mycryptopedia.com/blockchain-oracles-explained/*

In the first iteration of Haven and several subsequent designs since that time, the creation of a secure, accurate and high-performing oracle was considered key to the success of the protocol. However, since the creation and success of services such as Chainlink, which are designed purely to provide oracle functions as an independent data source, it is now clear that not only is a separate oracle not required to be built into the Haven system, but it is not desirable to do so. Doing so would increase centralization of the most important part of the conversion equation – pricing.

With this in mind, Haven Protocol has collaborated with Chainlink in order to utilize their oracle network for the processing and provision of pricing data. The Chainlink oracles for XHV/USD can be seen below

Source: *https://feeds.chain.link/xhv-usd*

Haven believes that it is vital to build in flexibility in pricing discovery from the start, and as such will not rely solely on one oracle system, but will be able to add, swap and remove oracles over time to ensure Haven uses best-in-class data now, and into the future.

Supply Scenarios

XHV is a pure Proof-of-Work (PoW) coin with the same emission curve as Monero, it has an initial minable supply of 18.4 million and a small tail emission once those 18.4 million coins have been mined.

This is a standard, well understood supply scenario in the cryptocurrency market. Now that Haven's offshore storage feature is live on mainnet, the above figures continue to apply to mining rewards, but no longer define the actual circulating supply of XHV since mint and burn will alter this dynamically as previously discussed.

In addition, once further xAssets (beyond xUSD) are live on the network, the circulating supply of XHV no longer defines the total market capitalisation of the Haven ecosystem. For this, it is necessary to consider the cumulative value of xAssets held as well as XHV itself.

This can be expressed as HNV or Haven Network Value and will be calculated as follows:

HNV = (XHV price * circulating supply) + xUSD circulating supply

Additional xAssets can easily be added into the calculation as they get added to the network.

To understand the potential future supply of XHV and the effect of that supply on the Haven ecosystem, the following high-level macro scenarios are presented.

Variables considered in these scenarios include:

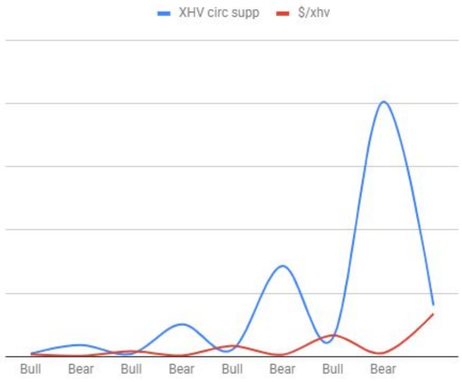
1. The increase in total market capitalisation in a market bull cycle (inc_Bull)
2. The decrease in total market capitalisation in a market bear cycle (dec_Bear)
3. The % of XHV coins sent to and stored in offshore at the end of a bull market cycle (perc_offBull)
4. The % of xAsset (using xUSD as an example) coins onshored back to XHV at the end of a bear market cycle (perc_onBear)
5. The % of the local ATH value of XHV within a bull cycle that is the average of all offshore transaction values (Eg. if the local ATH for XHV is $2.00 then 80% of that ATH is $1.60 and this would be the value used in these scenarios for offshoring if 80% is used for this variable) (perc_LATH)
6. The % of the local ATL value of XHV within a bear cycle that is the average of all onshore transaction values. (perc_LATL) §
    a. **Note: These values for 5 & 6 can be viewed as how accurate traders are when predicting tops and bottoms of markets.*
7. XHV volatility index - this value is used to simulate how correlated the volatility of XHV might be in comparison to Bitcoins volatility. A value of 1 being equal to BTC volatility, 0.5 being 'half as volatile', 2 being twice as volatile etc.

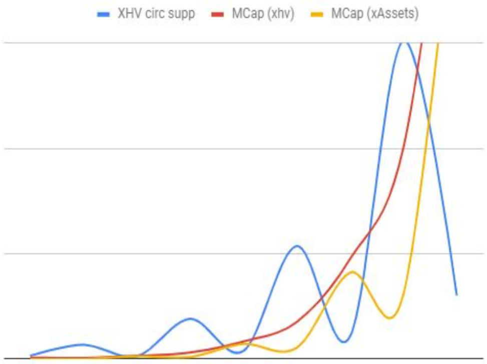
Scenario 1

Expansion in XHV Supply

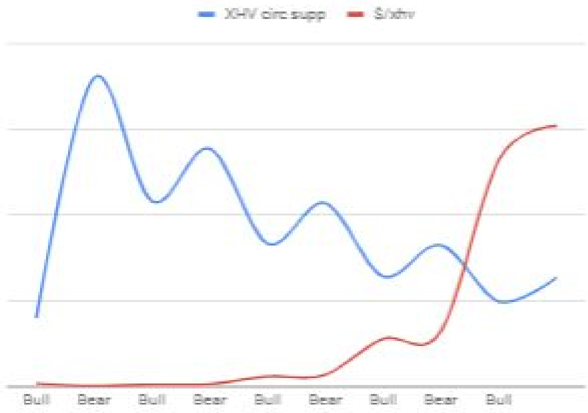
In this scenario we use values that will increase the supply of XHV in the market over time.

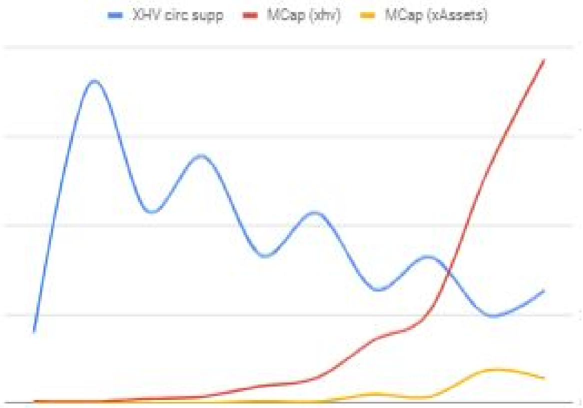
inc_Bull = 2500%
dec_Bear = 85%
perc_offBull = 80%
perc_onBear = 75%
perc_LATH = 90%
perc_LATL = 10%
iVol = 1.0





As can be seen in this model of extremely heavy offshore use and high trading accuracy, the use of offshore functionality in an expansion scenario keeps the price of XHV subdued, but over time increases market capitalisation of both XHV and the Haven ecosystem as a whole. This scenario is acceptable to the ecosystem since it lowers volatility of XHV price, which in turn alters the patterns shown and moves the scenario out of expansion, and into equilibrium (or even contraction) as can be seen in the charts below where the only change to the values used above is to iVol (0.5).



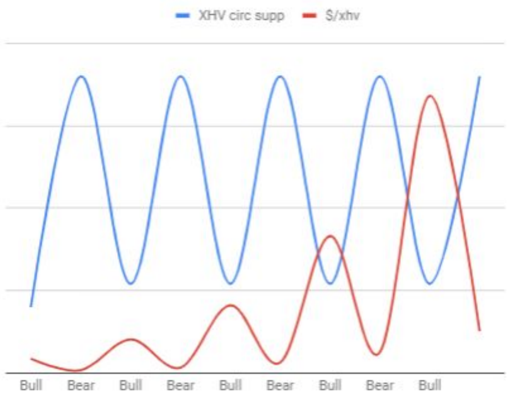


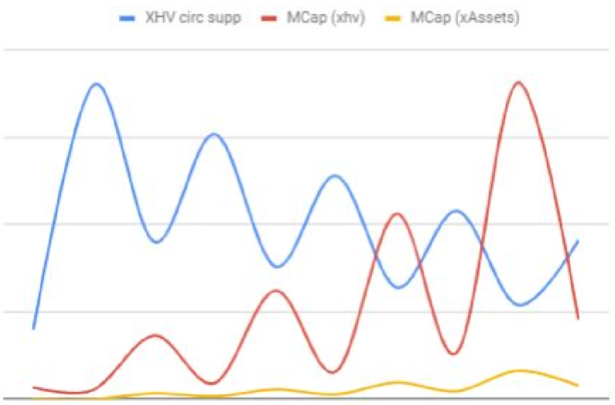
Scenario 2

Contraction in XHV Supply

In this scenario, values are used which deliberately create deflation in the circulating supply of XHV.

inc_Bull = 2500%
dec_Bear = 85%
perc_offBull = 50%
perc_onBear = 48%
perc_LATH = 60%
perc_LATL = 40%
iVol = 1.0





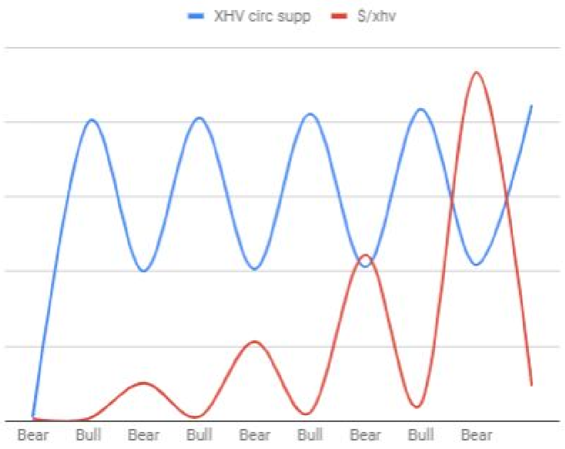
As can be seen in a contraction scenario, the price of XHV increases in volatility, creating the opposite effect from the expansion scenario over time and will move the pattern from contraction towards equilibrium or expansion.

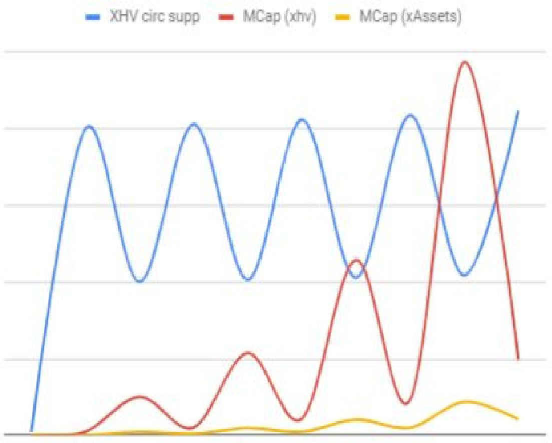
Scenario 3

Equilibrium in XHV Supply

In this scenario the prediction variables are set with medium use of offshore, and medium trading accuracy. As the central point between the two other scenarios, one can expect this scenario to play out repeatedly over time, with expansion and contraction scenarios both tending towards equilibrium.

inc_Bull = 2500%
dec_Bear = 85%
perc_offBull = 70%
perc_onBear = 50%
perc_LATH = 60%
perc_LATL = 40%
iVol = 1





In conclusion, while one cannot predict which scenario will play out at any given time, the protocol is designed to adapt to changing usage levels by expanding and contracting supply of XHV directly through user actions, creating a new and unique supply curve purely from natural and organic use.

Stability and Economics

Mint and burn requires little in order to implement in a basic form; just a known price at which to perform the conversion, and the ability to convert one type of asset to another on the same chain at that conversion rate.

To state the obvious, it is a very simple concept. That being said, the simplest of concepts are sometimes the hardest to fully understand, and to ensure that the Haven ecosystem uses a robust economic model, certain challenges must be addressed.

1. Supply transparency
2. Exchange based price manipulation
3. Proving and maintaining the value of synthetic assets in a PoW algorithmic ecosystem.
4. The Potential for a 'Run on the Bank' during periods of wider market volatility

These challenges will be addressed one at a time:

Supply Transparency

The original concept for Haven was based on having an unknown circulating supply of XHV and xAssets. The reasoning for this was to prevent manipulation of the network by large holders of XHV or xAssets.

After a great deal of consideration, community discussion, and consultation with expert advisors, it was decided that having a transparent circulating supply would actually be beneficial to the network in the following ways:

- It allows for more efficient monitoring of the Haven network, which means attempted attacks and large scale manipulation can be detected and mitigated much faster.
- It gives users greater confidence to enter the Haven network with the ability to view the number of XHV and xAssets in circulation at any given moment.
- It allows for greater visibility and therefore greater analysis on coin metrics websites. As a result, to ensure accuracy and visibility, each mint and burn transaction will be created in such a way that amounts will be discoverable through analysis of the blockchain, and displayed in the Haven block explorers. This will allow users to maintain standard Monero levels of anonymity and wallet address privacy while allowing a clear view of circulating supply.

Supply of each asset type is now visible and can be seen here:

https://explorer.havenprotocol.org/supply

Exchange Based Price Manipulation

Due to the nature of mint and burn, Haven's long standing promise that "1 xUSD will always be redeemable for $1 worth of XHV," and the price-smoothing action of moving averages within Haven's pricing system, certain measures are required to ensure that discrepancies between exchange prices and off/onshore conversions are minimized.

This minimization is performed by allowing a choice of transaction priority by the user. High priority transactions, with minimal unlock times, will be charged higher fees than low priority transactions with longer unlock times (where the fee will tend to near zero).

Since first launch, Haven contributors have been monitoring and analysing the data gained from activity over the first month of real world use. Since first launch, the original fee structure has been replaced by a far simpler and more stringent scheme to ensure network health short term while token distribution is with early holders. Over time, Haven envisions that fees and their structures will require revisiting and alteration to work alongside the maturity of the Haven network. The complete fee structure for the Haven network will be published alongside this paper, and maintained for reference at all times on the Haven Protocol website. https://havenprotocol.org/fees

One of the issues with many existing DeFi products is that you must have a particular token in your wallet in order to transact in another. This can cause unnecessary friction and costs just to use it.

Haven transactions overcome this by charging the fees in the currency being sent. This is shown in the table below:

Transaction Type	Fee Type	Fee payable in:
XHV transfer	standard tx fee	XHV
xUSD transfer	standard tx fee	xUSD
XHV -> xUSD exchange	exchange fees + standard tx fee	XHV
xUSD -> XHV exchange	exchange fees + standard tx fee	xUSD

Proving and Maintaining the Value of Synthetic Assets in a PoW Algorithmic Ecosystem

One of the biggest challenges of algorithmic synthetic assets, as well as one of the most frequently asked questions, is centered around the concept of "true value" or "source of value." Questions like "how can you claim xUSD is worth $1 when it has no collateral backing?" are asked often by users.

Once that question has been answered and understood (xUSD is "indirectly backed" by a varying and appropriate amount of XHV), users then focus on questions around the supply of, and liquidity of XHV itself. Since XHV supply will fluctuate due to offshore transactions as described above, both the supply expansion and contraction cases potentially change the dynamics of the entire ecosystem.

In all likelihood, taking into account the cyclic nature of cryptocurrency markets, the potential for both cases to arise is high. This is both expected and desirable. Fluctuations in circulating supply are absolutely required to allow for expansion and contraction in xUSD supply without creating ever larger volatility in the price of XHV.

The Potential for a ‘Run on the Bank’ During Periods of Wider Market Volatility

During rising market cycles (‘Bull markets’) in any commodity, traders often leave stable options in favour of volatile assets, and visa versa. With any traditionally ‘backed’ stablecoin such as USDT, the amount of backing is key to the stability of the backed cryptocurrency. Any deviation of ‘backed’ value from ‘market’ value creates a real danger to users, and creates a situation where there is a potential for unbacked value, and loss of peg to whatever asset the cryptocurrency is supposed to track.

Haven does not suffer this problem due to its use of mint & burn and coloured coins.

At all times, and in all situations a user can redeem 1 xUSD for $1 worth of XHV. This peg will never break.

Since Haven Protocol is implemented using a coloured coin model, it is capable of supporting not only xUSD, but also a range of other assets and commodities we call ‘xAssets’. This allows XHV itself to become the collateral for not only one, but a suite of private synthetic assets, extending the pegging mechanisms possible and turning the protocol into a platform with true use case and value to cryptocurrency users.

Who are the Haven team?

The Haven team is a community collective of developers and contributors and as such welcomes all input and contributions from any party.

The core development team is listed below.

Since taking over the management and development of the coin from the original developers, the community has benefitted from the continued support and guidance of several advisors, consultants and technology industry professionals who have made it their mission to fulfil the promise of Haven, and drive adoption of this vital part of the cryptocurrency landscape. The continued support and input from these individuals is greatly appreciated.

Core development team:

David Bandtock (@dweab) https://www.linkedin.com/in/david-bandtock-9647101/

David is a career technologist with a focus on product delivery and strategy, he has held senior positions in major UK Corporations and multiple technology startups over the past 20 years. With a background in Mathematics, encryption technology and Software development, David brings considerable experience both in technical delivery and large scale governance to Haven.

Neil Coggins (@neac) https://www.linkedin.com/in/neil-coggins-7972352/

Neil is a dedicated full stack software architect and developer. With over 20 years development experience in X86 Assembler, C++, Java, PHP and Javascript, Neil has spent the last 18 years designing and building cryptographic software.

@Marty (anonymous)

Marty is a front end developer with experience in a multitude of frameworks, and brings this to the fore with his work on the Haven wallets and websites.

@Pierre Lafitte (anonymous)

Pierre is a product design specialist, and creates all the user journeys and UI’s in the Haven product portfolio. Pierre is an experienced Front End crypto developer, is a long time contributor to Haven and will be leading the UX/UI side of development and bringing the UX visions of the team to reality.