# Bifröst Protocol: cross-chain bridges

## Enabling performant cross-chain bridges with full protocol security on THORChain.

devs@thorchain.org

July 2018

### Abstract

We propose an effective chain-agnostic bridge protocol that uses multi-signature accounts, cryptoeconomics and continuous liquidity pools (CLPs) to ensure security of assets traded across bridges on THORChain. This can be adapted for almost all major UTXO, account and contract-based assets including Bitcoin, Ethereum, their code-forks as well as their tokens. Validators are staked fully synced nodes that are part of the Validator Set for THORChain and secure the protocol. The Validator Set nominates a sub-set of Validators to form *k* bridges by being party to *m of n, n-1 of n* or *n of n* multi-signature accounts on the external chain. Each bridge offers different but observable security and performance characteristics to be selected by users to match their needs. The security of the bridge is *m* * *stake*, where *stake* is the stake held by each Validator in the sub-set. Incoming coins are locked in the multi-sig and minted as tCoins via CLPs on THORChain. Once minted, the tCoins are secured by the entire protocol on a single tokenChain and each tCoin exhibits verifiable fungibility despite being minted by different sub-sets on different bridges. THORChain tokens can also be safely deployed and recovered on any supported external chain via CLPs and the existing bridges. The CLP transmits a price for the tCoin which can be used by the protocol to measure the risk of each bridge; a risky bridge is one where the bridge security is less than the value of locked assets. The Validator Set will then re-shuffle a risky bridge, or move assets away to a more secure bridge; thus the bridges inherit the entire protocol's security. If a bridge is attacked, the Validators will slash the attackers and use slashed assets to restore the stolen coins, via on-chain governance and Foundation intervention.

# Table of Contents

# 1. Introduction

THORChain is an entirely new protocol being built to serve the needs of a liquid, instant and diversified future of value transfer. As it exists solely to connect all cryptocurrency assets with each other, cross-chain compatibility is a cornerstone feature of the protocol. THORChain has native features that allow it to become the desired ecosystem for all digital assets, and extremely easy to move assets on (and off) the network. Indeed, assets moved on to THORChain may be more valuable than their original counterparts due to a more favourable trading environment, lower fees and staking revenue opportunities available in THORChain, but not available on external chains.

## 1.1. Cross-chain Interoperability

The nature of blockchains is that truth is only preserved along a single chain, “longest chain” in Nakamoto Consensus, or in the case of Ethereum, “greediest heaviest observed sub-tree”, or GHOST. Naturally, any hard fork from that single chain will create a new and separate version of truth. Thus the world of many blockchains, each with their own version of truth and unique characteristics is a reality to grow accustomed to.

Moving information across chains in a multi-blockchain world is thus a necessary and highly-desirable part of the ecosystem. Copy and pasting information in a legacy database is trivial with the right access; but transferring information across immutable blockchains is unavoidably difficult. Not in the least is that both blockchains need to agree on the shared truth; but it is desirable to do in a trustless manner to be aligned with the fundamentals of the ecosystem itself. Most likely the immutable information that needs to be moved is value; and with value comes very strong incentives to intercept, double-spend and redirect that value.

There are a number of solutions to this problem that have been proposed and work somewhat, but most have to make tradeoffs between trust, security and performance. It is desirable to have a solution which is highly trustless, byzantine fault tolerant, is useable and has observable security.

## 1.2. Cross-chain Bridges

Cross-chain Bridges are simple; an asset is locked on one chain; whilst an identical asset is atomically “minted” on the other and sent to an address owned by the original party. The newly minted asset has fungibility with the original one by virtue of the fact that it can be used to redeem the original asset at the same ratio that it was minted. This newly minted asset represents the rights to unlock assets on the original chain; so as long as the bridge continues to exist without censorship or interference, then the tokenised asset can be handled as though it was the original asset. When the owner (anyone who has keys to spend) wishes to move back to the original chain, it is done via the same bridge; the asset is destroyed on the bridged chain and atomically unlocked on the original chain. Destruction can be done by provably burning the token (UTXO-based or account-based) or by deleting the token and decrementing the total supply of token if it is contract-based.

## 1.3. Atomicity

A fundamental part of the cross-chain bridge mechanism is that it must be atomic; a failed transfer must not cause a double-spend opportunity on the bridged chain. Time to finality on the host chain complicates this; as the bridge must wait on a threshold of finality before it mints new tokens. Typically this involves waiting for confirmations, such as six confirmations for Bitcoin and 24 for Ethereum.

Finality is influenced by blockchain re-organisation, where based on block propagation latency and the possibility that nodes may discover a longer chain or a GHOST, 0-conf transactions are not safe to work with. This drives exchanges and merchants to wait for confirmations that they feel safe with.

By virtue of what a blockchain is; finality is never 100% - there is always a possibility that the chain can be re-written with sufficient hash or staking power. There have been instances on smaller chains of re-writes despite as much as 50 confirmations with sufficient attack power. Typically the attacker will suffer an economic loss in attacking, but if the economic loss is less than the economic gain from a re-write then incentives favour the hacker.

A hard fork may also cause issues to the bridge in terms of what is considered to be the canonical chain. Tokens that have already been transferred to the bridged chain will have rights to spend both sets of coins in the forked chains, whether or not they have replay protection. Manual intervention may be required during times of hard fork events due to network uncertainty.

## 2. Existing Cross-chain Solutions

### 2.1. Rootstock 2WP[1]

Rootstock is an ethereum-compatible smart contract side-chain solution for Bitcoin. Rootstock have built a bridge that allows users to send Bitcoin to a multi-sig and have tokenised Bitcoin minted on the secondary chain. Rootstock use a concept of merge-mining and a drive-chain/side-chain implementation that gives primary custody to miners, with notaries in lieu of miner participation. It is preferred to have full miner participation in the consensus; however this is unlikely so a federation of notaries is employed to make up the security. Miner signalling is affected by the addition of an OP code that they insert in the coinbase. The presence of a federation is the primary drawback of this solution; parties must still trust a group of people to be honest.

### 2.2. Liquid Sidechain[2]

Liquid Sidechain is another Bitcoin sidechain solution that uses the concept of a Strong Federation to secure the bridge; essentially notaries in the same sense as Rootstock. It is not clear who can be a signatory, or if there is a limit to the number of notaries. This still does not escape the scourge of trusting a party.

### 2.3. POA Network Bridge[3]

POA Network is a sidechain to Ethereum with faster consensus; a Proof-of-Authority consensus algorithm where renowned members of the community put up their public reputation and identity to be block producers. The cross-chain bridge in this case is quite trivial as a multi-signature account tied to block producers is sufficient. Although efficient, the notion of having individuals control the bridge (and consensus) is a trustless tradeoff.

### 2.4. COSMOS Peg Zone[4]

COSMOS bridge the Ethereum chain with a peg zone that overlays a finality layer to Ethereum's probabilistic finality and convert the signature schemes between the two chains to make them compatible. The witness that signs the peg zone is a 2 / 3 consensus from the COSMOS Validator Set; so the bridge security matches the security of the entire COSMOS chain. This implementation does not make many compromises between trustlessness and efficiency. The only issue here is that it involves the full participation in the nodes that secure the COSMOS network, so performance characteristics are unknown.

### 2.5. Full Network Security

All pegs attempt to involve full network security in their bridges, however most concede defeat to trusted or semi-trusted bridges with less than full network security. The exception is Cosmos, which has full and trustless network security. Any less than full network security on a bridge means that a cartel always has the economic incentives to conduct a supermajority sybil attack and seize assets if they are valuable enough.

[1] https://uploads.strikinglycdn.com/files/ec5278f8-218c-407a-af3c-ab71a910246d/Drivechains_Sidechains_and_Hybrid_2-way_peg_Designs_R9.pdf

[2] https://blockstream.com/strong-federations.pdf

[3] https://poa.network/

[4] https://blog.cosmos.network/the-internet-of-blockchains-how-cosmos-does-interoperability-starting-with-the-ethereum-peg-zone-8744d4d2bc3f

# 3. Bifröst Protocol

## 3.1. Overview

The Bifröst Protocol is the solution for THORChain and combines full network security and oversight to cross-chain bridges with mechanisms to make bridges useable and performant. Additionally it requires no changes to the underlying blockchains that it forms bridges with. The bridges have compatibility with all major UTXO, Account and Contract-based cryptocurrencies.

The core Validator Set (100 staked Validators) are chosen to be signatories to multi-sig accounts to external chains. The external coin is moved into a multi-sig account, signed by the Validator Set. After observing finality, the Validator Set mints new tokens in THORChain via CLPs. The user can then exit via the same bridge. The token is destroyed via the CLP and then unlocked in the external multi-signature account to be sent to the user's address.

## 3.2. Considerations

There are a number of considerations that need to be addressed for this to work.

**Infrastructure.** The bridge requires all Validators to maintain fully synced nodes to all external chains. This involves a significant infrastructure overhead. Some chains may become out of sync from a crashed daemon, or some Validators may not have all chains synced at all. To subsidise the cost of infrastructure Validators are paid from the block rewards of the protocol. Each Validator must gossip the last block height from each chain and becoming out of sync by *n* blocks (determined by the protocol) causes the Validator to be penalised by being slashed. At some point of being out of sync of any of the chains by more than the specified *n* blocks, the Validator is evicted from the Validator Set. This ensures that all Validators in the Set are in sync in external chains and can process external transactions.

**Performance.** If there are 100 Validators in the Set, and 67/100 multi-signatures are required on external chains, performance will be a significant issue. For UTXO coins (especially ECDSA-based signature schemes) the bridge will be expensive, taxiing on the external network resources and soak up block sizes. For contract-based coins like ERC-20, the 67/100 multi-signature transaction will take over 67 blocks to execute and delay finality. For Cryptonote coins such as Monero and Loki where *m of n* multi-signature isn't supported, a *n-1 of n* or *n of n* multi-sig is required. Thus 99/100 or 100/100 will be next to impossible to orchestrate, even with full Validator Set participation. The solution is to reduce the size of the multi-sig required, but retain full Validator Set oversight, known as Supervised Subset Signatories (SSS) discussed in § 4.2.

**Redundant.** THORChain's Validator Set is designed to be flat and highly competitive to enter, so it is likely that Validators will be recycled regularly. Additionally a Validator may drop an external chain and be evicted, or may lose access to a private key and have a multi-sig compromised, as well as the risk associated with maintaining a *n-1 of n* multi-sig, the protocol must be redundant in how it maintains multi-sig accounts. The solution to this is Randomised Reshuffling Subsets (RRS), discussed in § 4.1.

**Fungibility.** It is imperative that all minted tokens have fungibility with original assets, as well as with each other. If a chain is bridged via multiple points then each token created from each bridge must be created from the same genesis on THORChain. The tokens must be so fungible, and the bridges must be so trustless, that tokens can entire via one bridge and exit via another with no loss in value. The solution to this is via use of THORChain's in-built CLPs. CLP interaction is discussed at length in § 5.

**Adaptable.** The protocol must be able to tolerate external chain changes, such as hard forks, chain re-orgs as well as adding new bridges to new chains at any time. Finally, the protocol needs to recover from the (unlikely) case of stolen assets from external accounts. The solution to this is effective on-chain governance with out-of-band community-sourced intervention. THORChain is built to be a self-amending ledger and the application of this that is relevant to the Bifröst Protocol is discussed in § 4.4.

## 4. Processing Bridge Transactions

There are a number of innovations proposed to solve the salient issues of bridges. The Bifröst Protocol includes concepts such as Randomised Reshuffling Subsets (RRS), Supervised Subset Signatories (SSS) and Signature Thresholds. This coupled with diversified bridges, trustlessly observable risk, underwritten assets and on-chain governance will see a mature implementation of the Bifröst Protocol the most effective in transmitting value across chains to and from THORChain.

### 4.1. Randomised Reshuffling Sub-sets

It is necessary to garner full protocol level security for bridges, but this comes at serious performance costs. The solution is ultimately a trade-off between security and performance as smaller signature requirements can be executed faster, but have higher risk. Using on-chain governance, *k* sub-sets of *n* signatures (such as 2 sub-sets with each 11 signatures) are voted to secure *k* Bridges. The Validator Set can use a VRF to nominate individual Validators to be signatories to each of the *k* sub-sets and *k* * *n* Validators are required to be sourced from the 100 staked validators. Every *x* blocks the multi-sig accounts are re-shuffled creating new multi-sig accounts.

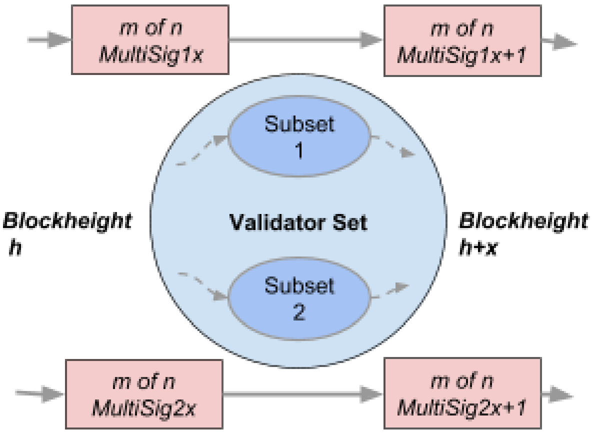


*Figure: k subsets, secured by m of n signatures from n validators, party to k mult-sigs. Every x blocks, a validator is swapped out and assets moved.*

Although there will be *k* multi-sig accounts on the external chain, the Validators mint tokens on the same internal tokenChain. Importantly, only one Validator from each sub-set is removed and replaced by a random external Validator, as this reduces the risk of locking out a multi-sig.

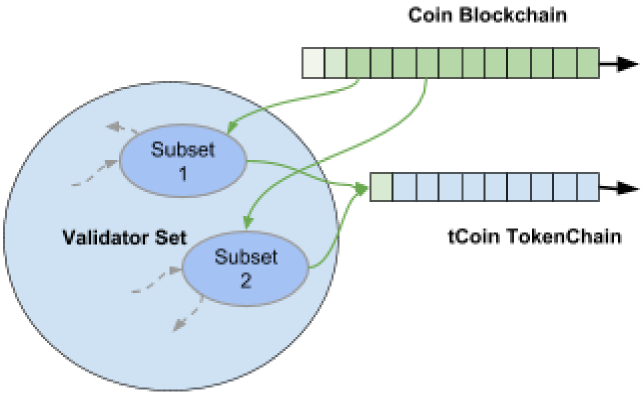


*Figure: Multiple bridges, one token. Sub-sets are re-shuffled regularly.*

The locked coins on the external chain will need to be moved as one transaction, signed by the parties required (*m of n, n-1 of n, n of n*, depending). The bridges can alternate between re-shuffling and signal ahead of time exactly when the re-shuffle will occur. After a bridge is re-shuffled it is a new multi-sig account; a consideration for users. If users send coins to a decommissioned bridge after it is re-shuffled (or during), there is a risk that not all Validators will be online or operating to honour the coin. This is especially true for *n of n* or *n-1 of n* multi-sigs, less so for *m of n*. The solution for this is building the correct user experience that communicates risks to users and showing the latest addresses. For contract-based coins there is the added benefit of using a proxy contract that will forward coins to the updated bridge. The Validators simply update the proxy contract at the same time as moving the coins. This proxy contract can be added in by on-chain governance.

**Diversification.** Multiple bridges can be set up deliberately with different security and performance characteristics for the end user's choice. A 3 / 4 Bridge is fast with low security, a 15 / 21 Bridge is slower but has more security, while a 20 / 21 Bridge would be the slowest, but with the highest security. As each bridge mints the same final tCoin which is secured by the entire protocol, users are simply making choices on the risk/performance of bridge transaction, as opposed to the security of the final tCoin. Entering on one bridge and leaving on another is also entirely feasible. This results in verifiable fungibility for each tCoin.

**Reliability.** By re-shuffling validators and moving assets every *x* blocks in multi-sigs the protocol and users can be assured that Validators are online and have access to their private keys. This will reduce the risk of loss of asset if more than one validator goes offline. Additionally, by moving assets regularly, *n-1 of n* or even *n of n* can be attempted depending on security required. Further if a Validator in a sub-set is evicted for other reasons, or confesses to a compromised or lost private key they can be removed in schedule by signalling.

### 4.2. Supervised Sub-set Signatories

It is necessary to supervise each sub-set with full protocol oversight to ensure that sub-sets are not victims to supermajority sybil attacks, where an attacker gains control of an entire sub-set to spend locked assets. The solution is for the Validator Set to observe each sub-set and take action when risks are higher than tolerated for bridges.

**Observing Risk**. The security of the bridge is thus equal to the sum of the stake for *m* validators in a *m of n* bridge if validators are completely slashed if they spend the locked assets. Thus rogue validators will observe the sum of locked assets and may consider an attack if they have more to gain by spending locked assets and being completely slashed. This is easy to observe and communicate to users. A bridge is risky when:

$$m * stake <= sum(locked\ assets)$$

By using trustless on-chain price feeds from CLPs the protocol can become self-aware of the risks of bridges. If the Validator Set observes the value of locked assets approaching the security of a bridge, they can do two things:

1. Spend a partial or full amount of the locked assets to a bridge with higher security, such as from the 3 / 4 bridge to the 15 / 21 bridge.
2. Re-shuffle the bridge to a multi-sig with a higher *m,* such as from a 20 / 21 to a 21 / 22 bridge.

The mechanism for this can be built into the consensus rules of the Validator Set, making the entire process trustless. This mechanism guards bridges that have partial security with full protocol security.

**Optimising Performance**. In the same way that multi-sig requirements for multi-sig accounts can be increased if the protocol observes that the security of a bridge is close to the value of locked assets, they can also be reduced. In the example that $1m of BTC is held in a 20 / 21 bridge, but the value of the stake is $20m, then the multi-sig requirements can be reduced to 10 / 11, where $10m is held. This will increase the performance of the bridge, but still have a factor of security over the value of assets. If the value of locked assets increases then in turn the signature requirements can increase.

### 4.3. Signature Thresholds

BLS signature thresholds can be used to further prevent issues during Validator Set re-orgs, as only a threshold of signatures is required to be achieved, instead of a specific aggregation of signatures. For example, if 15 / 21 signatures are required for a particular bridge, it does not need to specify the set of signatures required to form a 15 / 21 signature threshold, only that 67% is required. The reverse occurs to move the coin off the ecosystem. This translates to flexibility in who can be part of a sub-set, and tolerates validators leaving and re-entering the Validator Set, which could be a frequent occurrence in a healthy and competitive environment of validators. BLS signature threshold theory has been extensively researched by DFinity[5].

[5] https://dfinity.org/pdf-viewer/pdfs/viewer?file=../library/dfinity-consensus.pdf

### 4.4. On-chain Governance

On-chain governance, (known as Validator Signalling) is deeply integrated into THORChain and is a cornerstone of the protocol. Validator Signalling is a perpetual and continuous process by which Validators signal support for on-chain proposals and integrate them. This process is outlined in detail in the Validator Signalling whitepaper[6]. With effective on-chain governance the protocol can quickly adapt to a changing environment. This is especially required for managing cross-chain compatibility and its inherent risks.

**Underwriting Assets**. Locked assets can be underwritten with the value of the stake of the Validators that secure the bridge, allowing a recovery (full or partial) of stolen assets. If a bridge suffers a supermajority attack, the rogue validators will be slashed immediately by the Validator Set. The seized stake can then be used to liquidate on markets to restore the stolen assets. Firstly Validators must signal to freeze the slashed assets, then they need to be moved to a community or Foundation wallet. If community approval is gained the Foundation or Trustee can purchase assets to replace stolen assets to restore full fungibility.

**Chain Support**. All aspects of supporting external chains such as their selection and support, hard forks, finality thresholds, multi-sig addresses, bridge number and thresholds as well as proxy and token creation contracts can be administered using on-chain governance.

[6] Validator Signalling Whitepaper

# 5. Implementation

## 5.1. Entering a Bridge

When a Bridge is first established using Validator Signaling, the tCoin and tokenChain is created by the Validator Set using a Genesis transaction (GenTX) with 0 supply. Each time a new coin enters via the bridge, the supply of the GenAcc in the CLP is updated accordingly by the Validator Set, and the tCoin is sent to the recipient. The GenAcc is filled with liquidity from either the project team, Foundation or self-interested liquidity stakers. GenTX, CLPs and Liquidity mechanisms are explained in the THORChain whitepaper[7].

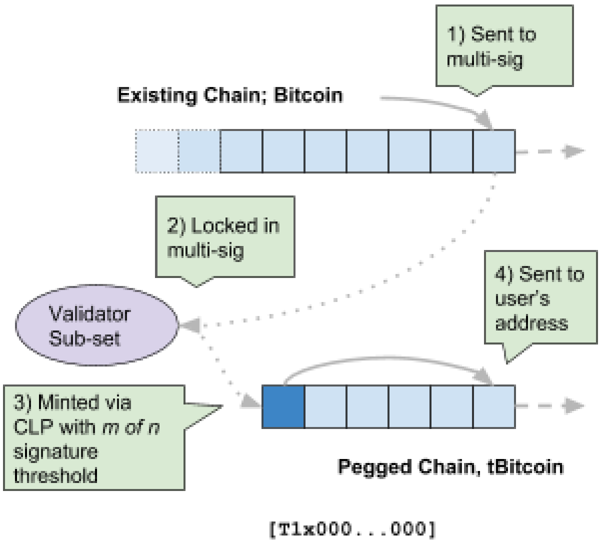
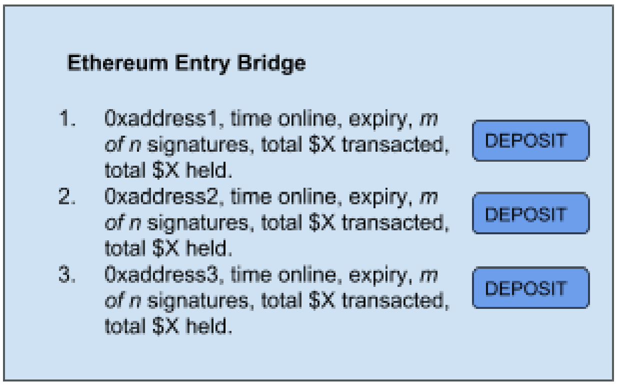


*Figure: Sending coins to THORChain.*

The tokenIndex is chosen as simply the next index available and the information is stored in the token GenAcc. In the case that Bitcoin is tokenised onto the T1 chain and Ethereum is tokenised onto T2, the GenAcc would store (simplified):

```
{T1:([BTC multi-sig addr1, expiry], [BTC multi-sig addr2, expiry]), etc}
```

This allows the Validator Set to update the external multi-signatures if they ever need to change, such as if they were compromised. The user experience is simple and trustless, with an overview similar to the following:



Ethereum Entry Bridge

1. 0xaddress1, time online, expiry, *m of n* signatures, total $X transacted, total $X held.
2. 0xaddress2, time online, expiry, *m of n* signatures, total $X transacted, total $X held.
3. 0xaddress3, time online, expiry, *m of n* signatures, total $X transacted, total $X held.

*Figure: Enter Bridge User Interface, queried from the GenAcc.*

Importantly the data can be verified at any time by reading the GenAcc, so the bridge interface can be displayed on any front-end. As infrastructure is subsidised by block rewards on THORChain the bridges can operate feelessly.

Contract-based cryptocurrencies such as ERC-20, NEP-5 and QRC-20 can also be sent on to THORChain using the same parent multi-signature address. The GenACC stores the token's root contract address alongside the sequential tokenIndex. In this way, anyone can create TokenChains for contract-based cryptocurrencies and achieve deconfliction. The following would be the contract-based implementation:

```
{T7:(Parent tokenChain (T2), contractAddr);

T8:(Parent tokenChain (T2), contractAddr);

T9: etc}
```

This allows anyone to send any token via its parent bridge to THORChain. If already created, it is simply added to the existing chain. If it hasn't been created a new tokenchain is added.

[7] THORChain Whitepaper

### 5.2. Trading tCoin

Once inside the ecosystem tCoins can be transacted freely and fungibly. These include all trades and transactions types. If traders can observe that their tCoins have 1:1 asset backing and that original assets can be claimed at any time, then tCoins will be 1:1 to Coins. A running total of locked assets as well as minted assets would be publicly available and verifiable, allowing anyone to see full backing. If traders can observe the security of a bridge in economic value and know that at any stage a loss of locked assets would be under-written by slashed validator stakes, then they are further reassured of full fungibility of assets.

Since bridges don't have to charge bridge fees, and transaction fees on THORChain will be substantially lower than the original chains, the tCoins *may* have a higher perceived value than original assets. Take for example December 2017, where Bitcoin fees were greater than $50 for a sustained period of time[8]. Transacting on Bitcoin would erode value, whilst transacting a tBitcoin on THORChain would save a considerable amount in fees. Further, assets on THORChain can be staked in CLPs to earn on liquidity fees, with no inflation risk.

These characteristics are important to ensure that tCoins >= Coins. If this is achieved then CLPs will attract self-interested arbitrageurs to correct market slips and maintain tCoin >= Coin prices. THORChain can also offer a more attractive trading environment by using Foundation assets to purchase Bitcoin off the market, establish bridges and lock tBitcoin in the CLPs. This will effectively bootstrap the on-chain liquidity for the tCoin. Once bootstrapped self-interested stakers can add further liquidity to earn on liquidity fees, growing on-chain liquidity. These mechanisms are described in the THORChain Whitepaper[9].

### 5.3. Exiting a Bridge

tCoins can be exited at any time by simply using the Bridge in reverse. The tCoin is sent to the CLP in a special transaction which destroys it and updates total supply by the Validator Sub-set. Once block finality is achieved ~1 second, then the Validators can sign the external multi-sig to release the assets to the user's address. Signing the external multi-sig and the finality of the transaction is limited by external chain characteristics.

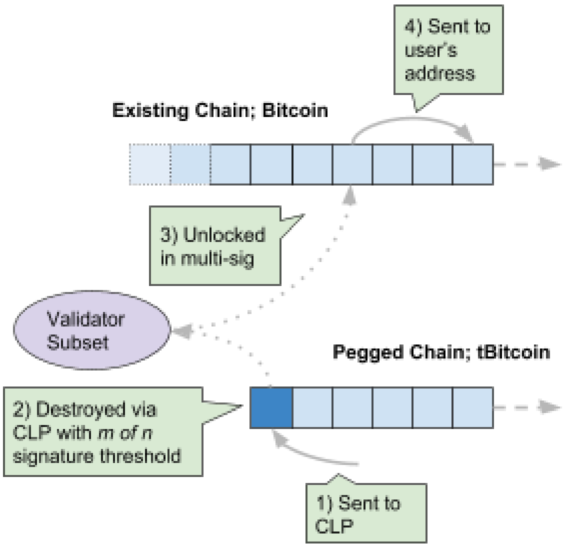
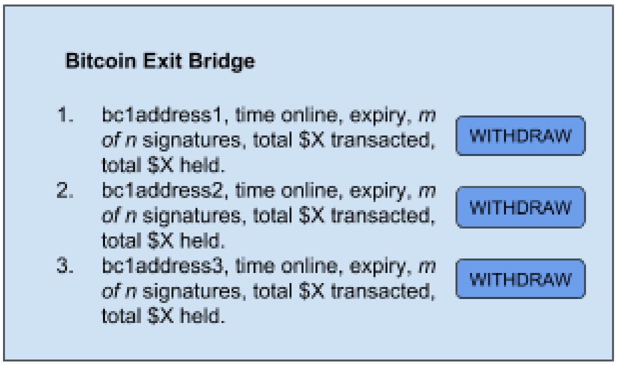


*Figure: Sending coins out of THORChain.*

As with entering the bridge, users can choose from various bridges with different security and performance characteristics. On-chain or client side checks can ensure that a user does not exit a bridge with assets that will exceed the security of a bridge. Again, the value of the assets can be known on-chain by the CLP pricing.



**Bitcoin Exit Bridge**

1. bc1address1, time online, expiry, *m of n* signatures, total $X transacted, total $X held.
2. bc1address2, time online, expiry, *m of n* signatures, total $X transacted, total $X held.
3. bc1address3, time online, expiry, *m of n* signatures, total $X transacted, total $X held.

*Figure: Exit Bridge User Interface, queried from the GenACC.*

Token exits can also be performed, even tokens that have never been created on external chains.

[8] https://www.blockchain.com/charts/transaction-fees-usd

[9] THORChain Whitepaper

This can be done by allowing the Validator Set to be party to a token creating account, that simply deploys a token standard with variable supply. The owner of the contract is the multi-sig for the bridge. Each time a new token is sent, the multi-sig updates the supply in the token contract and forwards the tokens to the user address.

Users must specify which external chain they want to exit on, as it is their choice. As an extension of this, since the "original" asset is the THORChain asset which has a supply controlled by THORChain Validator Set, tokens can be exited onto multiple external chains, and still retain fungibility. A Rune could have 1:1 representative coin on Ethereum, as well as EOS and NEO. The THORChain MerkleChain DHT tracks assets and their linked chains.

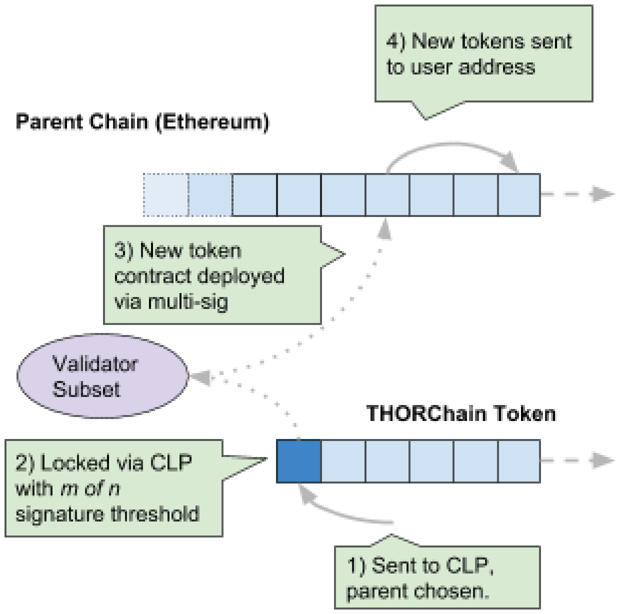

4) New tokens sent to user address
Parent Chain (Ethereum)
3) New token contract deployed via multi-sig
Validator Subset
THORChain Token
2) Locked via CLP with *m of n* signature threshold
1) Sent to CLP, parent chosen.

*Figure: A THORChain token is exited and created on a parent chain.*

Having cross-chain compatibility is imperative for THORChain to augment the entire ecosystem by bringing the benefits of decentralised trading and liquidity to all existing tokens. With the correct design, THORChain may be a far more favourable environment for digital assets to exist and be the ecosystem of choice for the creation of digital assets.

# 6. Bifröst CLPs

## 6.1. Generating a Bifröst CLP

Continuous Liquidity Pools (CLPs) are a cornerstone of THORChain, generating on-chain liquidity, transmitting prices to the protocol, providing price anchors to the Flash Network, storing auditable token information and allowing fees to be paid in any token. The CLPs in the context of the Bifröst Protocol are used to mint, destroy, lock and unlock tCoins.

The Bifröst CLPs are a variant of variable supply tokens, explained in the THORChain whitepaper. The GenAcc stores the following information for a Bifröst CLP:

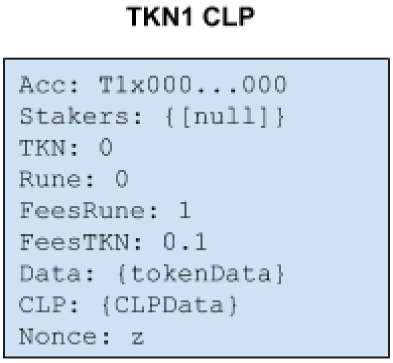
| | | |
|---|---|---|
| **Token** | `T1` | TokenIndex |
| **Ticker** | `BTC` | Ticker to display |
| **Name** | `Bitcoin` | Name to display |
| **Supply** | `100` | Total Circulating |
| **Decimals** | `8` | Decimals |
| **Reserve** | `1.0` | Fractional reserve of the CLP. |
| **Owner** | `SELF` | The Owner is the protocol, and can only be changed by the Validator Set. |
| **Bridges** | `{[multi-sig addr1, expiry, minSig, numSig]; etc}` | Bridge Addresses with expiry times to re-shuffle. and *m of n* signature thresholds. Multiple bridges can be set. |

*Table: A Bifröst Token CLP; here for Bitcoin.*

## 6.2. CLP Transactions

There are a number of CLP transactions relevant to the Bifröst protocol, more than the basic transactions covered in the THORChain Whitepaper. The following is an overview.

**Creating.** Using Validator Signalling a new bridge is nominated, synced and set up with the correct information. An altruistic party (the Foundation) or a self-interested forward-thinking party (Project Team), will then bootstrap the chain by sending in the first liquidity deposit to create the CLP liquidity. There will be two liquidity deposits required, the Rune, then the tCoin.



TKN1 CLP

```
Acc: T1x000...000
Stakers: {[null]}
TKN: 0
Rune: 0
FeesRune: 1
FeesTKN: 0.1
Data: {tokenData}
CLP: {CLPData}
Nonce: z
```

*Table: GenAcc for T1*

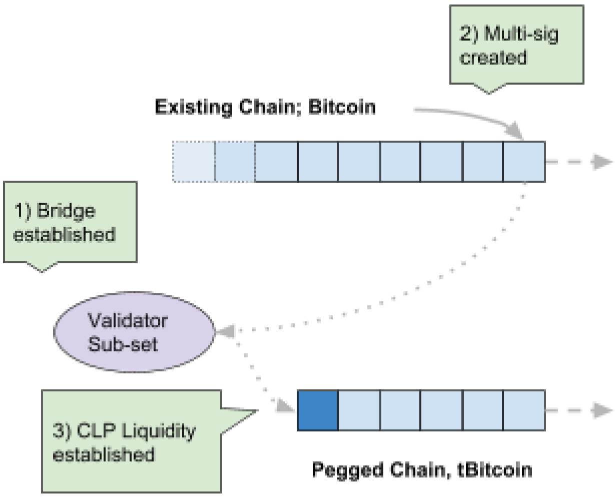


*Figure: Bootstrapping a new tokenChain*

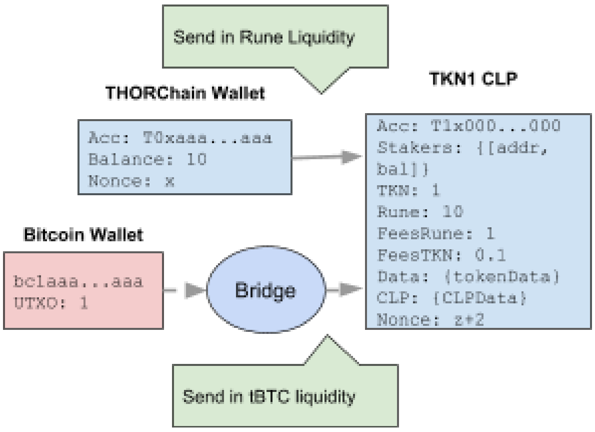


*Figure: Creating liquidity in the CLP.*

**Minting.** Every time a new coin enters the bridge the Validator Sub-set then performs the following transaction on the linked CLP:

```
mint(balance, address)
```

The CLP will then:

1. Create an additional amount of coins
2. Immediately send them to the address
3. Update the total supply in the tokenData field

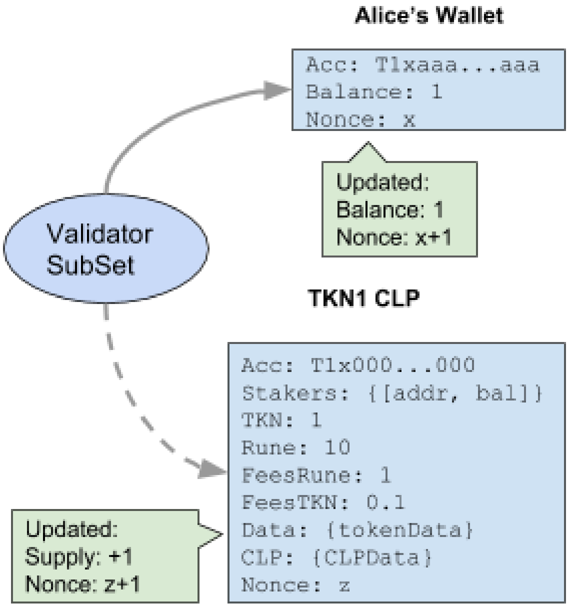


*Figure: Minting new tokens via CLP.*

**Destroying.** Every time a coin exits the bridge the Validator Sub-set then performs the following transaction on the linked CLP:

```
destroy(balance, address)
```

The CLP will then:

1. Destroy the amount of coins from user address
2. Update the total supply in the tokenData field
3. Unlock tokens from the external multi-sig
4. Send to user address

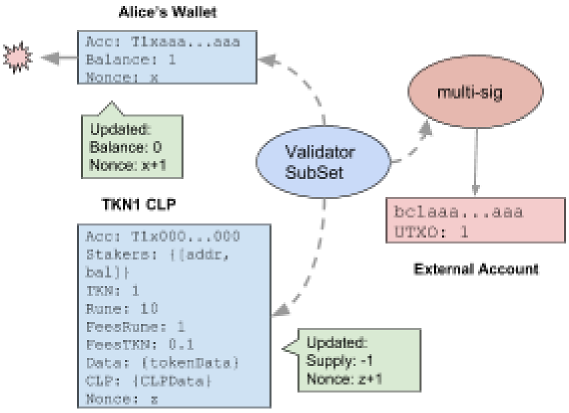


*Figure: Minting new tokens via CLP.*

**External Deployment.** A user can deploy their THORChain tokens to an external chain:

```
deploy(balance,     address,
parent)
```

The Validator Set will then:

1. Destroy the amount of coins from user address.
2. Deploy an external token contract with the correct balance on the parent Chain.
3. Send to user address.

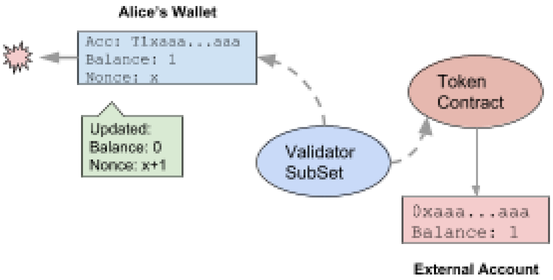


*Figure: Deploying new tokens via CLP.*

**External Recovery.** A user can recover their THORChain tokens from an external chain:

```
recover(balance,     address,
parent)
```

The Validator will then:

1. Destroy the amount of coins from external wallet.
2. Update the token contract.
3. Unlock tokens from the CLP to the user's address.

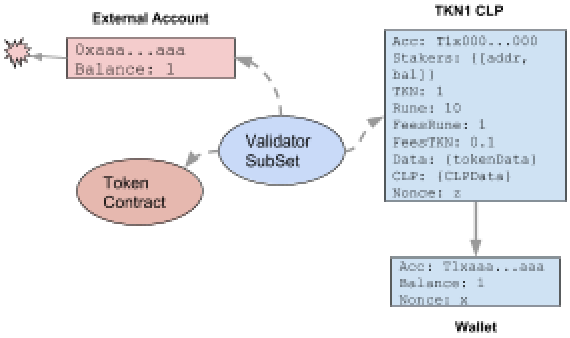


*Figure: Recovering tokens via CLP.*

# 7. Security

## 7.1. Overview

The Bifröst Protocol is designed with specific regard to attack vectors, and makes effort to mitigate risk with protocol security, cryptoeconomics and game theoretical approaches. Attack vectors can come from external chains, inside THORChain, inside the sub-sets and the most significant; a supermajority sybil attack on the protocol itself.

In this analysis the specific example to highlight and frame the discussion is when $100m in external assets are locked in multi-sigs, and the protocol's Validator Sets have less than $150m in Rune staked; such that 67% = $100m.

## 7.2. External Lone-ranger Attacks

**External Assets.** Any assets locked in an external multi-sig account become a honeypot for attackers and incentivise attacks, let alone anonymous multi-sigs which is the proposition of the Bifröst Protocol. The private keys to secure the multi-sigs are held by specialised validators that hold stake in the THORChain ecosystem, not the external chain. Thus for an external attacker to double-spend or attack a multi-sig; the security threshold is unequivocally the same as the security of the entire chain linked. Chains with small hash power have historically been attacked, and indeed the attack cost is known ahead of time with services such as Crypto51[10]; a tool to calculate the cost of 51% attacks on various chains. In this case an attacker would attempt to gain 51% network power for as long as they could re-write the chain for more than the number of confirmations that exchanges permit in the disposal of assets; typically 6 confirmations. This is around an hour of control for Bitcoin. By requiring internal stake, external attack vectors are subservient to the characteristics of external chains and this is a sufficient mitigation of risk.

**Internal Assets.** Assets held in THORChain are only vulnerable to internal attacks as the permissioned validators that control internal assets are always internal to THORChain. There is no attack vector here.

## 7.3. Internal Lone-ranger Attacks

**External Assets.** Any assets locked in an external multi-sig account are controlled by permissioned validators. An agent would need to become permissioned before they could attack.

**Internal Assets.** Assets locked in internal accounts, such as CLPs, are secured by the protocol. An attacker would need to exploit a vulnerability in the protocol to siphon assets from CLPs and this would be patently obvious once the first attack occurred. An attacker would need to become permissioned as a validator to exploit other attack vectors.

## 7.4. Validator Attacks

As part of THORChain's design all Validators hold stake in the network, the amount set by the lowest bidder. An annual proposed inflation of 5% motivates holders to become Validators, or delegate tokens to Validators. Validators are incentivised to watch each other for rogue activity in order to slash an attacker's stake and earn it for themselves. Thus an attacking validator must always consider full loss of stake as the economic cost for an attack.

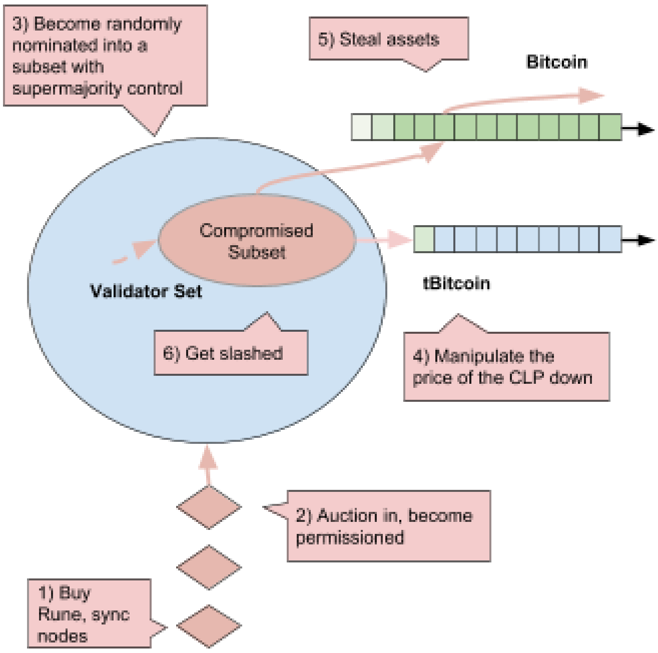
**Single Validator.** Assuming the protocol works as designed with byzantine resistance, a single rogue validator is effectively risking their entire stake for an attack that will never be effective. They may attempt a double spend, but it is assumed a single honest validator (from a pool of 100) would notice and publish the proof of attack to slash the attacker. With correct protocol design the attack vector here is limited to the validator making an attack and getting away with it for as long as it not noticed. This can be mitigated by enforcing an unbonding period before a validator can reclaim their stake from resigning or being evicted as a validator. THORChain's proposal of 14 days unbonding period is a sufficient period of time to allow retrospective auditing of blocks by the community and validators and impose slashing rules.

[10] https://www.crypto51.app/

**Validator Sub-set.** Validator sub-sets are nominated to control the keys to external chains, as well as aggregating signatures for signature thresholds for internal signing of CLP transactions. The value proposition of the Bifröst Protocol is that offers bridges with performance, whilst still retaining the security of the entire protocol. A key mechanism is that the protocol can observe the risk of a bridge by using prices from CLPs, and take action to reduce risk, or increase security. The inherent risks are a supermajority sybil attack in a sub-set. The protocol attempts to mitigate this risk by randomly appointing who can be part of a sub-set, as well as recycling validators every *x* blocks. The following are considerations for an attacker who can achieve supermajority in a subset, but not supermajority of the protocol:

1. A cartel need to first become permissioned into the Validator Set, which involves buying into the network and becoming fully-synced and compliant full nodes. Block rewards whilst being compliant would compensate the infrastructure cost.
2. The cartel would need to then manipulate the CLP pricing of the chosen asset down in order to prevent the protocol from taking corrective action. Depending on the volume of the asset, this would incur a high arbitrage cost to them. The cartel would need to perform this for at least *x* blocks.
3. The cartel require the value of the locked asset (whilst they manipulate it) to increase in value to be higher than their combined stake; as their stake will ultimately be staked. Since they are manipulating the price of the asset down in (2), it would incentive traders to buy the cheap asset and send it off the network to sell at higher prices elsewhere. This would actually reduce the amount of locked asset in the external accounts, removing the honeypot.
4. If (3) does not work in time, the cartel would then need to rely on probability to be nominated such that they control a supermajority in any sub-set.
5. A cartel, who by chance, gains the supermajority in a subset has a limited amount of time to attack, *x* blocks, before they are likely to lose the edge.
6. Assuming (4), the cartel would then need to spend the external assets prior to (5) and hide the act for the duration of the the unbonding period.
7. If their act is discovered inside the unbonding period, then they are slashed. This is highly likely. Their economic gain is thus:

*(stolenAssets) - ((slashedRune) + (arbitrageCost * x blocks))*



3) Become randomly nominated into a subset with supermajority control
5) Steal assets
Bitcoin
Compromised Subset
Validator Set
tBitcoin
6) Get slashed
4) Manipulate the price of the CLP down
2) Auction in, become permissioned
1) Buy Rune, sync nodes

*Figure: The attack path to steal external assets*

Manipulating CLP pricing in order reduce the perceived cost of an asset in order to allow an attack to occur is covered in the THORChain Whitepaper.

### 7.5. Network Attacks

**Supermajority Attacks.** The largest risk to THORChain is a protocol level supermajority sybil attack with an attacker gaining 67% control of all validators. At 67% control everything can become compromised, including external assets, internal assets, protocol rules and the validators themselves.

Arguably the cost of attaining 67% of the Validators would be high and likely to cause a runaway price action on the Rune as 67% of all bonded Runes would be bought by a single attacker. An attacker may instead try to attain the support of 67% of the nodes to form a cartel, but again this is unlikely since the nodes are self-interested and have an ongoing infrastructure cost that needs to be compensated. Attacked THORChain would likely result in a devaluation of the network.

**Protocol Attacks.** An attacker may instead try to create an asymmetric attack vector by bringing in changes to the protocol via on-chain governance and Validator Signalling. The counter to this is creating a well-designed governance system with empowered minority mechanisms who can meaningfully influence voting; as minorities may have more awareness of long-range protocol attacks through crowd-intelligence. Validator Signalling is discussed at length in the On-chain Governance Whitepaper.

## 8. Conclusion

We have proposed an effective chain-agnostic bridge protocol that uses full THORChain network security to allow assets to be moved seamlessly on and off the ecosystem. By randomly reshuffling the validators used in multi-signature accounts and having multiple bridges with different security and performance metrics, each bridge maintains optimal performance whilst being supervised with full protocol security. If a validator attempts to spend locked assets they are slashed and the seized assets can be used to restore the assets. With the use of CLP price feeds the protocol can become self-aware of the risks of each bridge, and make adjustments to signature requirements.

Tokens on THORChain gain low fees, on-chain liquidity, and a superior trading environment. All assets can be used by traders to stake inside CLPs to earn on liquidity fees, so tCoins will likely be more valuable than original assets. Tokens created on THORChain can also be easily deployed on multiple other chains but recovered easily.

The Bifröst Protocol will be built for THORChain to augment the entire cryptocurrency ecosystem, but can also be readily deployed to other protocols that have the key requirements to support it.

## 9. References

1. Anon, n.d. POA Network. [online] POA Network: public Ethereum sidechain with Proof of Authority consensus by independent validators. Available at: <https://poa.network/>
2. Anon, n.d. Total Transaction Fees in USD. [online] Blockchain. Available at: <https://www.blockchain.com/charts/transaction-fees-usd>
3. Unchained, C., 2018. The Technicals of Interoperability-Introducing the Ethereum Peg Zone. [online] Cosmos Blog. Available at: <https://blog.cosmos.network/the-internet-of-blockchains-how-cosmos-does-interoperability-starting-with-the-ethereum-peg-zone-8744d4d2bc3f>