ElasticBridge: Trustless Cross-Chain Bridge for Transferring Rebase Tokens

Timothy Yalugin
timauthx@gmail.com

November 29, 2021

Abstract

This paper proposes an approach for secure and robust bridging of ERC-20 compatible rebase tokens between EVM-based source chain and WASM-based receiver chain deployed on Polkadot network.

1 Introduction

Emerging applications of blockchain technology indicate an ongoing trend for market diversification. Once the idea of sovereignty-preserving application-specific chains became feasible, it quickly displace a single *chain-maximalism* dogma. We are now live in a world where *multi-chain* networks like Polkadot [**?**] and Cosmos [**?**] are gathering diverse interconnected applications around their relay-chains and hub-chains respectively, creating *the Internet of Blockchains*[1].

Frameworks like Parity's Substrate[2] and Tendermint's Cosmos SDK[3] streamlined the process of building and deploying sovereign and flexible blockchains, allowing them to have different consensus mechanisms, finalities, state models, governance, etc.

1.1 Cross-chain Bridges

Effectively improved on scalability by sharding on-chain data and parallelizing transaction execution, we encountered new challenges with cross-chain interoperability. Many newborn ecosystems, all with unique features and value propositions, found themselves near of Tower of Babel[4] — unable to communicate and effectively cooperate to provide meaningful services together.

The natural reaction was the development of cross-chain bridges — protocols that allow two (or more) exogenous chains to share state by the means of inter-chain transaction passing. We have seen various implementations of these protocols ranging from trusted federations to truly trust-less decentralized Relayed SPV solutions like the IBC Protocol [**?**].

1.2 Rebase Tokens

Meanwhile, cryptocurrencies — the original use case of the Distributed Ledger Technology (DLT), also do not show any signs of innovation slowdown. Projects like Stellar [**?**] commit to building a foundation for transferring digital currency to fiat money across borders. Controversial privacy-oriented currencies like Monero [**?**] allow completely untraceable payments. Finally, fiat-pegged tokens like Tether [**?**] and multi-collateralized stablecoins like MakerDAO's Dai [**?**] have been one of many attempts to tackle the price-volatility — issue often synonymous with digital currencies.

Rebasing — a novel technique applied for synthetic commodities for achieving practical price stability by algorithmically adjusting circulating tokens amount, thus translating price-volatility into supply-volatility. Rebase tokens (or price-elastic tokens) demonstrated low correlation not only with stocks, fiat currencies, and precious metals, but with other cryptocurrencies, most notably with Bitcoin.

While the low correlation aspect attracts investors seeking to diversify their portfolios, the unique volatility pattern of rebase currencies makes them valuable building block[5] for DeFi as

[1] Notion proposed by Cosmos developers. https://cosmos.network
[2] The blockchain framework targeting Polkadot multi-chain network. https://substrate.io
[3] Framework of development zone-blockchains for Cosmos network. https://tendermint.com/sdk
[4] Idiom that references mythical narrative when people lose ability to speak same language.
[5] Claim from Ampleforth documentation. https://ampleforth.org/economics

well. For example, the longest-running algorithmic stablecoins based on rebasing — Ampleforth [**?**] constructs its protocol with dependence on fast actors who benefit from short-term trades. By leveraging certain windows in time during rebase to make lucrative trades, these actors propagate adjustments of supply made by protocol back into the price.

1.3 Outlined challenges

Notably, supply expansions and contractions — actions performed by rebase protocol to adjust total supply, affect token amount universally: all wallet balances, treasury, and accounts reserved for smart contracts. Last introduces a significant consideration for cross-chain bridges design since it generally relies on escrow account where dispatched tokens would remain locked on the source chain.

Assuming that bridge protocol will mint wrapped representation of original token on the target chain, we can draw a corresponding conclusion that *without changes to original bridging protocol wrapped token supply will eventually become out of sync with rebase-driven native token supply.* The inconsistency between amounts of native and wrapped tokens would then lead to non-deterministic behavior, mostly obvious — *translation conflicts when transferring rebase tokens back to the source chain.* This would also expose both chains to novel attack vectors that would not otherwise be possible.

1.4 Proposal statement

The purpose of this paper is to firstly shed light on the implications that elastic supply of dispatchable tokens would have on cross-chain interoperability protocols applied to them. We will then go through key considerations that should be taken into account while designing a bridge for transferring such tokens. The Paper will center around the protocol proposal for the bridge connecting EVM-compatible source chain and Substrate-based parachain on the Polkadot network. However, the proposed solution should effectively work on chains of other origins as well. Finally, we will finish current paper with follow-up questions and ideas for future exploration.

2 Design consideration

General method by which cross-chain bridges operate can be described as **lock-and-mint/burn-and-release**[6]. Bridge would lock the tokens in the source chain (SC) and mint a wrapped representation onto receiver chain (RC). Reverse process would then be to burn wrapped tokens from RC and unlock original tokens on SC. The described approach allows to *preserve circulating token supply by distributing it across both chains.* This property will also propagate with each following transfer in N-Way bridge, where tokens can be passed downwards to another "satellite" chains without going back to the source chain, as long as contract of burning wrapped tokens on leaving "satellite" chains will be kept.

The given overview of the conventional bridging protocol allows us to distinguish two states, where we should consider exceptions when such protocol is applied to rebase tokens. These states are 1) forward movement from source chain to target chain 2) backward movement back to source chain. In the upcoming section we will go though each of them.

2.1 Forward bridge movement

Native tokens (NT) locked on the source chain escrow account would act as collateral for newly minted wrapped tokens (WT) on the receiver chain. Since WT are pegged by NT on 1:1 basis their price ratio will initially have the same relation as well. We are now able to add new assumptions to the table:

1. As long as WT are pegged with same amount of NT the price ratio will remain unchanged
2. If after rebasing on SC there will not be an equal adjustment of NT supply, then the amount of WT become out of sync with NT so as their prices
3. If supply adjustments made by rebase protocol will propagate to RC, then the amount of NT and WT will remain equal so as their price[7]

[6] https://chainbridge.chainsafe.io/live-evm-bridge
[7] with rebase reaction lag

Basing on defined assumptions we can immediately outline first and foremost consideration that directly translates to protocol scope requirement — *in order to preserve price and supply synchrony it is required to ensure that rebasing rules of source chain are followed on all receiver chains.*

Reasons why price-ratio and supply-ratio of wrapped and native tokens must remain equal will be covered in following sections. For now please assume that *since wrapped token is by definition a direct representation of original token we generally want their price to be synced so as the supply.*

We also need to consider possible negative implications of locking tokens to the rebase process itself. Consider the formula for calculating supply adjustment coefficient used in Ampleforth's Supply Policy Contract [**?**] when current Ample[8] price is outside the threshold of the target price:

$$supplyDelta = \frac{currentPrice - targetPrice}{targetPrice} * totalSupply$$

Since formula takes into account exactly the total supply, which does include amount of tokens locked in the bridge's escrow account, we can in fact justify that dispatch of rebase tokens from their source chain is possible. The opposite would be true for circulating supply which will decrease by the number of locked tokens.

2.2 Backward bridge movement

By the means of locking dispatchable tokens in an internal escrow account, bridge protocol ensures its ability to redeem the required amount of tokens back without changing total supply when minting new ones. The obvious problem with rebase currencies bridging is the fact that since the escrow account balance is a variable out of the bridge's full control, we cannot guarantee the fulfillment of the redeeming contract. We will address this issue as *translation conflict* later on.

Since translation conflict is type of non-deterministic behaviour that is caused by the deviations occurring between rebase-driven native token supply and relatively static[9] wrapped token supply, we can re-use previously defined assumptions to conclude — *similarly how the price of wrapped tokens relies on the collateral supply synchrony, the bridge protocol rely on it to fulfill its redeeming contract and thus it must be preserved.*

2.3 Price-Supply equilibrium

The core process of rebase protocol is to actively seek a price-supply equilibrium — state when current token supply constitutes into the target price through exchange-rate deviations caused by fast actors. This contract is what defines rebase currencies value to potential investors, traders, and other type of DeFi actors. Assuming so, the case with wrapped token that represents rebase currency but do not comply with its supply adjustment rules would probably make no meaningful sense for all actors besides those that are either malicious or driven by greed with similar aftermath.

We can foresee a potential attack vector derived from the native-to-wrapped token supply deviations caused in chains that lack rebasing support: *malicious fast actor can transfer to non-rebasing chain tokens, bought during expansion phases when price is lower, in order to redeem them after contraction phase when price is higher.* This gives malicious actor a clear advantage over honest actors since account balance on non-rebasing chain will not contract whereas all source chain balances will.

2.4 Archiving supply synchrony

Let us now define possible approaches of structuring multi-chain version of rebase protocol. These are the ones that will be outlined by the current paper:

a. Rely on source chain to govern rebase calculation and then on protocol to propagate adjustments to each satellite chains

b. Let each satellite chain govern its own wrapped token supply individually, yet with respect to source chain defined rules[10]

[8] Ampleforth protocol native token

[9] Wrapped token supply is driven only by bridging protocol

[10] Multi-chain governance of rebase protocol is also possible but will be outside of the scope of this paper

First approach implies to having a single relay-chain that by knowing total token supply both domestically and outside could effectively rebase on it for achieving price-supply equilibrium universally by broadcasting supply delta to all satellite chains. From the perspective of the relay-chain all satellite chains are light clients with certain account balance that must be equally rebased.

Second approach attempts to bring more decentralization to rebase governance by partitioning supply adjustments across all multi-chain participants. Although at the first glance this method does seem more advantageous, it has some caveats we have to consider first. Assuming that each individual chain will track its local total supply to perform rebase calculation, bridge by minting new tokens will introduce an additional factor that is not necessarily considered by the rebase protocol. We can of course argue that since bridge being a de facto sole-producer of such tokens on satellite chains it would also act as a rebase governor, but the described issue goes further into economical scope — *addition or reduction of tokens by means of minting or burning would most certainly alter the price-supply equilibrium established since the last rebase recurrence.* Furthermore, since we cannot deterministically assume that supply information will be equally propagated to price by each individual group of fast actors trading different wrapped token representations, there is a prominent possibility that *as soon as the underlying prices diverge to the extent of target price threshold, supply deviation will most certainly happen too.*

Presumably, it still is possible to get over the described issues with some novel layer zero consensus mechanism, yet considering that use of central relay-chain for multi-chain interoperability has proved its robustness and security, the second approach will probably be an overhead.

3 Protocol implementation

ElasticBridge — is a proposed implementation of cross-chain bridge connecting EVM and WASM chains for exchanging rebase tokens in a decentrilized, secure and trustless fashion. The base chain that holds a ERC-20 token Contract is deployed on Ethermint network, that EVM-compatible, Proof-of-Stake and build with Cosmos SDK framework. On the other side of the bridge we got application-specific chain with WASM-based runtime powered by Substrate framework and deployed on Kusama[11] network.

Building a robust, secure, and trustless bridge requires a decentralized governance structures implemented through solid technology solutions along with a well-balanced incentive model based on an elaborate game theory. Thus, similarly to building a blockchain from scratch, bridging a few of them without any foundation is an immensely difficult task. That is why ElasticBridge will rely on existing solutions.

3.1 Toolchain

At the time of writing, there two prominent candidates for a EVM-to-Substrate bridge development frameworks to consider: ChainBridge[12] and SnowBridge[13]. Although current version of the ChainBridge operates under a trusted federation model and the SnowBridge is claimed to be trustless SPV solutions, the more likely choice out of two would still be ChainBridge, which according to ChainSafe, the developers behind the project, is a planned to move to the trustless model as well. The developers of Ampleforth currency made a similar decision for their AMPL Bridge, which is powered by Meter Passport[14].

Additionally the fact that Ethermint is a part of Cosmos ecosystem allows us use to utilize IBC protocol as well. Projects that implemented IBC for Substrate are QuantumTunnel[15] and `substrate-ibc` pallet[16].

4 Conclusion

4.1 Follow-up questions

[11] Biggest test network on Polkadot ecosystem, that is oriented on rapid and radical innovation. https://kusama.network

[12] https://chainbridge.chainsafe.io

[13] https://snowbridge-docs.snowfork.com

[14] N-Way bridge that is build with ChainBridge framework. https://docs.meter.io/passport

[15] https://github.com/ChorusOne/quantum-tunnel

[16] https://github.com/octopus-network/substrate-ibc