# Cross-chain Secure Messaging

**jistro.eth 0xVato.stark ariutokintumi.eth**

Last update: April 20, 2024

## ABSTRACT

In this study, we explored solutions, and challenges in cross-chain implementations, providing a glimpse into the strengths and limitations of different of cross-chain solutions.

Keywords: Cross-chain, Blockchain

## CONTENTS



## 1 CROSS-CHAIN SOLUTIONS

### 1.1 Axelar

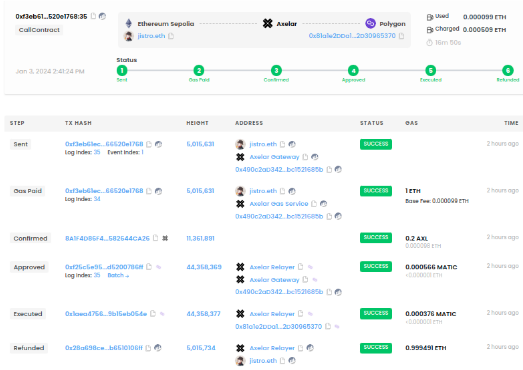
In the case of Axelar, we reference the documentation for calling a contract on chains [1]. The sender employs the `gateway.callContract` command, specifing the message, `destinationChain` (a string indicating the chain name), `destinationAddress` (a string denoting the Smart Contract address of the receiver), and the value in the local token of the chain, covering the transaction cost within Axelar.

The corresponding code can be analyzed and utilized from the following link: `https://github.com/Roll-a-Mate/Research/blob/main/0003-CrossChainSecureMessaging/Code/Axelar.sol`

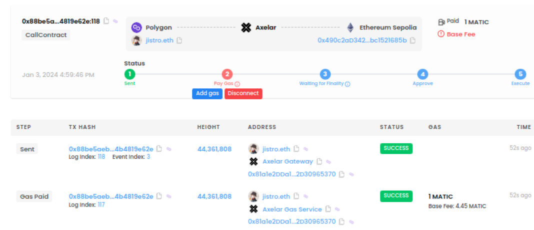
Upon sending the message in this instance, from Ethereum on the Sepolia chain with the contract address `0x490c2aD342C42fb60148d57cAEEB6Cbc1521685b` to Polygon Mumbai using the contract address `0x81a1e2DDa1aE7DFA35631075898ee72D30965370`, obtaining the transaction hash is crucial for monitoring the process (see Figure 1a). The entire duration of this transaction is 16 minutes and 50 seconds, incurring a cost of 0.000509 ETH, equivalent to 1.12 USD (as of January 3, 2024).

Unfortunately, the interaction from Polygon Mumbai to Ethereum Sepolia failed due to the higher base fee, approximately 6 Matic tokens (as of January 3, 2024), resulting in the message not being sent (see Figure 1b).

Axelar has the most straightforward and developer-friendly implementation. However, concerns arise about the security using strings for `destinationChain` and `destinationAddress`, and the volatile base fee for chains that do not have ETH as their main token, making cost estimation a bit challenging.



**(a)** Transaction from Ethereum Sepolia to Polygon Mumbai



**(b)** Failed Transaction from Polygon Mumbai to Ethereum Sepolia

**Figure 1.** Axelar Transactions

### 1.2 Hyperlane Implementation

During the implementation of Hyperlane, we encountered challenges with the smart contract integration. The documentation proved to be confusing and, in some instances, contained errors. Consequently, developers spent additional time investigating a task that could have been completed more efficiently.

Once the challenges were overcome in the implementation phase, we created a Smart Contract on Ethereum Sepolia with the address `0xECBc8094DebB430d5E75afCAFe6a18422f73BF32`. This Smart Contract is designed to make a call to a receiver contract in Polygon Mumbai with the Smart Contract Address `0x69D69632dfC69C1714CED0CE2CE6F15285da6478`.

The corresponding code can be analyzed and utilized from the following link: https://github.com/Roll-a-Mate/Research/blob/main/0003-CrossChainSecureMessaging/Code/Hyperlane.sol

Using the transaction hash in the explorer provides the result of the Cross-Chain call process, taking 57 seconds. Unfortunately, the explorer does not provide the transaction cost in ETH for the transaction (see Figure 2).

While seeking assistance in the official Discord channel, the team and I discovered some bugs in the documentation, particularly regarding the `IMailbox` smart contract invocation. There were misspelled variables for the `dispatch` function. Hyperlane assured us that they intend to improve this documentation as soon as possible.

The lack of comprehensive documentation and absence of pricing information on the explorer for Hyperlane posed significant challenges during implementation. The investment of additional time was necessary to ensure a correct and efficient integration process.

### 1.3 Chainlink CCIP

Utilizing the documentation for sending arbitrary data [2], the sender employs the `outer.ccipSend` function. This function facilitates the transmission of a message, requiring parameters such as `destinationChainSelector` (a uint64 indicating the chain location), `_receiver` (an address value denoting the Smart Contract address of the receiver), and the fee, which can be covered in LINK tokens or the native chain token.

The corresponding code can be analyzed and utilized from the following link: https://github.com/Roll-a-Mate/Research/blob/main/0003-CrossChainSecureMessaging/Code/CCIP.sol

Executing this operation from the contract `0x76c9b90A6c62c24FA711d4aF123D8FA3f52D6c58` in Ethereum Sepolia with a deposit of 5 LINK tokens to the contract, the message is sent to Polygon Mumbai using the contract address `0xa810b54E2c5c89Db6069Bc14613Af86586142aE7`. In Axelar, the transaction hash plays a crucial role in obtaining cross-chain transaction details. The entire duration of this transaction (see Figure 3) is 21 minutes, incurring a cost of 0.045014064168071695 LINK, equivalent to 0.64 USD (as of January 3, 2024).


**Figure 2.** Transaction Result for Hyperlane Cross-Chain Call

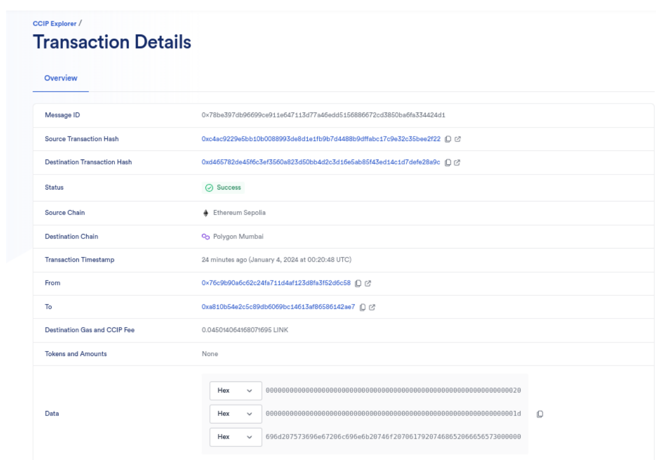



**Figure 3.** CCIP transaction

Chainlink's CCIP brings a similar experience to Axelar, but the price can be stable if we use the LINK token as a transaction fee. The problem with this solution is the time required for validation from Ethereum to any chain, which is significantly slower compared to other solutions.

## 2 LAYER ZERO

Based on the documentation from Layer Zero Getting Started [3], we first defined the endpoint address and the endpoint id. An interesting aspect of the endpoint (especially for the testnet chains) is that almost all chains share the same address, providing a seamless developer experience where every number corresponds to a uint32.

For this implementation, we deployed a smart contract for the Ethereum Sepolia testnet with the address `0x3E6a9bFA458242b6E123700244485169a31E48B7` as the sender. The smart contract receiver is deployed on the Optimism testnet with the address `0x58B5baB0a2df888501b8bC4CEF2391Df002674A5`. This choice is made because the Message Execution Options are optimized for messages sent to the Optimism testnet. Citing the LayerZero documentation for Message Execution Options [4]:

> ”LayerZero provides robust Message Execution Options, allowing you to specify arbitrary logic as part of the message transaction, such as the gas amount and `msg.value` the Executor pays for message delivery, the order of message execution, or dropping an amount of gas to a destination address.”

The corresponding code can be analyzed and utilized from the following link: https://github.com/Roll-a-Mate/Research/blob/main/0003-CrossChainSecureMessaging/Code/LayerZero

When executing the sender contract using 1 ETH for testing the Layer Zero refund function, the entire duration of this transaction (see Figure 4) was 25 seconds. We incurred a cost of 200,000 Wei (0.0000000000002 ETH), which, as of the writing of this section (January 12, 2024), is equivalent to 0.00000000051 USD.

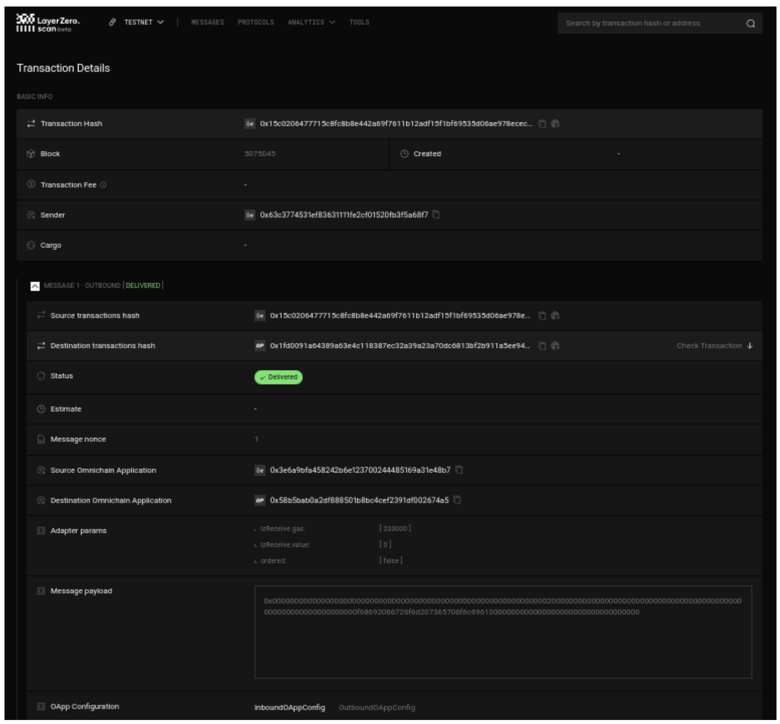



**Figure 4.** Layer Zero ETH to Optimism Transaction

Layer Zero provides a permissionless protocol for sending messages chain to chain. However, as observed in Hyperlane and Axelar, the transaction cost for every transaction can be volatile because Layer Zero uses the main token as the transaction cost.

| Solution | Delay Ethereum ⟶ Polygon | Delay Polygon ⟶ Ethereum | Cost per Message Ethereum ⟶ Polygon | Cost per Message Polygon ⟶ Ethereum | Consensus |
|---|---|---|---|---|---|
| Axelar | 18m 52s [5] | 16m 39s [6] | 0.000595 ETH [5] | 12.17 MATIC [6] | Proof of stake nodes |
| Hyperlane | 3m 3s [7] | 10m 20s [8] | 0.00002339021594448 ETH [7] | 20.380072606431995 MATIC [8] | Per-origin-chain [9] |
| CCIP | 25m 46s [10] | 3m 6s[11] | 0.008620804720114726 LINK [10] | 2.534953023940595 LINK [11] | Chainlink Oracles [12] |
| Layer Zero | 3m 46s [13] | 20m 40s [14] | 0.000352500782360452 ETH [13] | 0.006596250005936625 MATIC [14] | Oracles [15] |

**Table 1.** Comparative Cross-Chain Solutions Data using Ethereum and Polygon.

## REFERENCES


[1] Axelar, "Axelar Documentation." [Online]. Available: https://docs.axelar.dev/dev/general-message-passing/gmp-messages

[2] "Send Arbitrary data." [Online]. Available: https://docs.chain.link/ccip/tutorials/send-arbitrary-data

[3] Layer Zero, "Getting started," Jan. 2024. [Online]. Available: https://docs.layerzero.network/contracts/getting-started

[4] ——, "Message execution options," Jan. 2024. [Online]. Available: https://docs.layerzero.network/contracts/options

[5] "InterchainTransfer Ethereum to Polygon." [Online]. Available: https://axelarscan.io/gmp/0x8f09ee1516b4ee6c138f8adfc2f0fdb0572edfdfcbc4b12070a6a6dd698342c7:42

[6] "InterchainTransfer Polygon to Ethereum." [Online]. Available: https://axelarscan.io/gmp/0x879fd7a3197dd361473dba4442d6e1a86975d885c7267a0baeced15a2b42e4b0:258

[7] "Hyperlane Ethereum to Polygon." [Online]. Available: https://explorer.hyperlane.xyz/message/0x9727b24c169fa48db962212427e85cd86671b25deabe9d81c17e4539f1a48402

[8] "Hyperlane Polygon to Ethereum." [Online]. Available: https://explorer.hyperlane.xyz/message/0xa402225b029b583415533f381c4dd21668f8ad0157119a0e9cd6e0916bdb318a

[9] Hyperlane, "Validators." [Online]. Available: https://v3.hyperlane.xyz/docs/operate/validators/run-validators

[10] "Ethereum to Polygon." [Online]. Available: https://ccip.chain.link/msg/0x204a1e1cc5ca41222cb343f8cc9e8903b717503359824fd0a5743b4f96ea0ee9

[11] "Polygon Mumbai to Ethereum Sepolia." [Online]. Available: https://ccip.chain.link/msg/0xb610268e55bb709cf1ee7c68cf4a25ad0a033582ce24cd10af965b9fd9ab44bd

[12] Chainlink, "CCIP Off chain Components." [Online]. Available: https://docs.chain.link/ccip/architecture#off-chain-components

[13] "Transaction Ethereum to Polygon." [Online]. Available: https://layerzeroscan.com/tx/0x5d375296994774a8131916cadb541ec0e770b10ae1b84bcf4f3ee1f4b24f4de2

[14] "Transaction Polygon to Ethereum." [Online]. Available: https://layerzeroscan.com/tx/0xdfe90973de7eea233bd5ff8ef4d962a5ec21c1d2d4cb335ff39f245c18411b79

[15] S. Obasi, "LayerZero: The First Omnichain Interoperability Protocol for Cross-Chain Communication," Jun. 2023. [Online]. Available: https://medium.com/coinmonks/layerzero-the-first-omnichain-interoperability-protocol-for-cross-chain-communication-e5d5e37b99a9