Message Verification

We recommend checking out theWhat is a Bridge? section if you haven't already! The most important component of any crosschain messaging system is how messages areverified across domains. There are a number of different verification methods,each with their own tradeoffs .

Summary Table

External Optimistic Local Native Rollups Mechanism External actors attest message correctness. Messages incur 30 min delay and are disputable by Watchers. Messages are verified by the parties who they affect. E.g. a swap is verified by both counterparties. Messages are verified directly by chain validator set, typically using a light client + using ZKPs for scalability. Special case: L1 runs a "full node" directly verifying state transitions (not just consensus!) of rollup. Trust assumptions Fully assumes honesty of verifier set. PoS systems have better security than MPC or Oracle/Relay ones. Assumes that at least 1 Watcher is online, honest, and able to send a transaction to chain. Assumes liveness of all parties within a certain period to verify message. Light clients verifyconsensus but not state - assume honest relayer can give them the correct block to verify. Reasonable/minimal assumptions around L1 censorability and ZK proof schema. Tradeoffs Most trusted option compared to underlying chains. (1) Message latency - can be combined with local verification to cut latency in many cases. (2) Censorship risk of Watchers - can be mitigated with economic incentives. Doesn't really scale beyond 2 parties. This means onlysome types of messages are supported - see Authentication for details. Difficult to build + needs custom work for each chain. Doesn't really work for rollups. Proving state transitions only works for rollups, not between two separate chains. Examples LayerZero, Celer, Multichain, Axelar Nomad, Hyperlane, Hop (also local), Across (also local) Connext v0, Hop (also optimistic), Across (also optimistic) IBC, Succinct Labs, zkBridge, zkIBC Optimism, Arbitrum, ZkSync, Starkware

External

Externally verified bridges rely on an 3rd-party set of actors to attest the correctness of data that is transported between chains. This is typically represented as an multisig, MPC system, PoS validator set, or oracle/relay.

External verification is easy to extend to any chain, and can support generalized messages at low latency. However, the security of a message passed between chainsfully relies on the verifier set - this means that security scales with the size of the set. In most cases, externally verified bridges have far less security than the underlying chains, implying that they are not trust-minimized.

External bridges with large validator sets and staked incentives (e.g. PoS systems) are more secure than their smaller counterparts.

Optimistic

Optimistic bridges usefraud proofs to ensure the validity of data relayed across chains. Every message that passes through an optimistic bridge remains in a “pending” state during the dispute window until it is considered valid. During this time,watchers can dispute the message if the data is incorrect.

Optimistic bridges assume that there is at least one Watcher online somewhere (1-of-n assumption) that can send a transaction to chain. They also assume that Watchers will not abuse their position to censor the system. Censorship risk can be minimized using economic (dis)incentives.

Native

In a natively verified bridge, a chain's own validator set verifies information coming in from another chain. This happens by implementing alight client of one chain within the VM of another, typically within a zk-snark to keep expensive computation offchain.

Because light clients verifyconsensus but not the state transitions of the block itself (the latter requiresData Availability), they assume the existence of anhonest relayer who cangive them the "correct" block from a connected chain. This makes them vulnerable to attacks where the validators of one chain produce an invalid block (for example, through a 51% attack) to spoof a message through the bridge.

This honest relayer assumption of light client based bridges is the key point behind Vitalik's often-discussedMultichain not Crosschain post. Natively verified bridges have an additional tradeoff in that they require custom work for each chain with a new consensus mechanism. For some domains, for example with rollups, there may not be a light-client based strategy for bridging that works without additional trust tradeoffs.

Despite the above, natively verified bridges (regardless of whether they use ZKPs) are the best mechanism for moving data between two discrete blockchains.

Special Case: Local

Locally verified bridges are a special case of bridging where only the parties involved in a given cross-domain interaction verify the interaction. This is typically done through schemes like atomic swaps.

The challenge of locally verified bridges is that they require the liveness of all parties involved in a transaction, degrading UX and making it impractical to execute transactions that involve more than 2 parties. This further means that only some types of transactions are supported. For example, a peer-to-peer transfer is possible because there are only two parties involved, but many types of peer-to-contract interactions would not be possible asevery chain user would be involved.

Local verification techniques are typically used in conjunction with other mechanisms such as optimistic bridges to improve latency. You can learn more about what types of messages can be executed immediately through local verification in our Authentication section.

Special Case: Rollups

Rollups are another special case. Rollups exist specifically to solve for the "honest relayer" assumption that exists with light client based bridges. They do this by creating a strict heirarchy between the parent chain (the L1) and the rollup, with all rollup data being available on the L1. This allows the L1 to run a "full client", directly verifying the state transitions of the rollup itself.

Rollup bridges arethe most secure mechanism to communicate between any two domains. Period.

However, they can only exist between a rollup and its L1. See our section on Intercluster vs Intracluster communication for details.

Previous Modular Bridges Next Use Cases Last updated9 months ago On this page *Summary Table * External * Optimistic * Native * Special Case: Local * Special Case: Rollups

Edit on GitHub