LI.FI Security Review

AllBridgeFacet.sol(v2.1.0)

Security Researcher

Sujith Somraaj (somraajsujith@gmail.com)

Report prepared by: Sujith Somraaj

July 22, 2025

Contents

1 About Researcher

Sujith Somraaj is a distinguished security researcher and protocol engineer with over eight years of comprehensive experience in the Web3 ecosystem.

In addition to working as a Security researcher at Spearbit, Sujith is also the security researcher and advisor for leading bridge protocol LI.FI and also is a former founding engineer and current CISO at Superform, a yield aggregator with over $170M in TVL.

Sujith has experience working with protocols / funds including Layerzero, Edge Capital, Berachain, Optimism, Sonic, Monad, Blast, ZkSync, Decent, Drips, SuperSushi Samurai, DistrictOne, Omni-X, Centrifuge, Superform-V2, Tea.xyz, Paintswap, Bitcorn, Sweep n' Flip, Byzantine Finance, Variational Finance, Satsbridge, Earthfast and Angles

Learn more about Sujith on sujithsomraaj.xyz or on cantina.xyz

2 Disclaimer

Note that this security audit is not designed to replace functional tests required before any software release, and does not give any warranties on finding all possible security issues of that given smart contract(s) or blockchain software. i.e., the evaluation result does not guarantee against a hack (or) the non existence of any further findings of security issues. As one audit-based assessment cannot be considered comprehensive, I always recommend proceeding with several audits and a public bug bounty program to ensure the security of smart contract(s). Lastly, the security audit is not an investment advice.

This review is done independently by the reviewer and is not entitled to any of the security agencies the researcher worked / may work with.

3 Scope

- src/Facets/AllBridgeFacet.sol(v2.1.0), src/Facets/ChainflipFacet.sol(v1.0.1), src/Facets/DeBridgeDlnFacet.sol(v1.0.1)
- src/Facets/MayanFacet.sol(v1.2.1), src/Facets/RelayFacet.sol(v1.0.1)
- src/Helpers/LiFiData.sol(v1.0.0), src/Helpers/SwapperV2.sol(v1.0.1)
- src/Interfaces/ILiFi.sol(v1.0.1), src/Libraries/LibAsset.sol(v2.1.1), src/Periphery/ReceiverChainflip.sol(v1.0.1)

4 Risk classification

Severity level	Impact: High	Impact: Medium	Impact: Low
Likelihood: high	Critical	High	Medium
Likelihood: medium	High	Medium	Low
Likelihood: low	Medium	Low	Low

4.1 Impact

High leads to a loss of a significant portion (>10%) of assets in the protocol, or significant harm to a majority of users.

Medium global losses <10% or losses to only a subset of users, but still unacceptable.

Low losses will be annoying but bearable — applies to things like griefing attacks that can be easily repaired or even gas inefficiencies.

4.2 Likelihood

High almost certain to happen, easy to perform, or not easy but highly incentivized

Medium only conditionally possible or incentivized, but still relatively likely

Low requires stars to align, or little-to-no incentive

4.3 Action required for severity levels

Critical Must fix as soon as possible (if already deployed)

High Must fix (before deployment if not already deployed)

Medium Should fix

Low Could fix

5 Executive Summary

Over the course of 3 hours in total, LI.FI engaged with the researcher to audit the contracts described in section 3 of this document ("scope").

In this period of time a total of 7 issues were found. This review focussed only on the changes made from the previous version, not the code on its entirety.

Project Summary	
Project Name	LI.FI
Repository	lifinance/contracts
Commit	30b38a1
Audit Timeline	July 17, 2025 - July 18, 2025
Methods	Manual Review
Documentation	High
Test Coverage	High

Issues Found	
Critical Risk	0
High Risk	0
Medium Risk	0
Low Risk	1
Gas Optimizations	0
Informational	6
Total Issues	**7**

6 Findings

6.1 Low Risk

6.1.1 Overpaid fees are lost

Context: AllBridgeFacet.sol#L172

Description: The `_allBridgeData.fees` parameter is not validated, so any excess fees paid are lost because the AllBridge router does not refund overpaid fees.

Recommendation: Consider implementing basic validations as follows:

```
uint256 fees = ALLBRIDGE.getTransactionCost(5) +
  ALLBRIDGE_ROUTER.getMessageCost(
     5,
     IAllBridge.MessengerProtocol.Allbridge
  );

if(_allBridgeData.fees > fees) _allBridgeData.fees = fees;
```

LI.FI: Acknowledged. Added more documentation around the behavior in d354cfa0129198d80d47f4ba6f70f05871a214a1

Researcher: Acknowledged.

6.2 Informational

6.2.1 Destination chain id in AllBridgeData is redundant

Context: AllBridgeFacet.sol#L60, AllBridgeFacet.sol#L130

Description: The current AllBridge facet accepts two different chain ids from the user, one in AllBridgeData and other in the BridgeData and validate if both are equal. Instead, the function could solely rely only on the destination chain id in bridge data using teh conversion function, which can reduce code complexity and gas usage.

Recommendation: Consider removing destinationChainId from AllBridgeData and cast the destinationChainId from BridgeData for usage inside the `_startBridge()` function.

LI.FI: Fixed in 8fa11e4658f4fbc70e5154f29e18999cf1382809

Researcher: Verified fix

6.2.2 Inconsistent type

Context: AllBridgeFacet.sol#L130-L133, AllBridgeFacet.sol#L191

Description: The function `_getAllBridgeChainId()` silently casts the **uint32** all bridge chain ids into **uint256** chain id exhibiting inconsistency.

Recommendation: Either declare all bridge chain IDs as uint256 instead of uint32 in the constants.

LI.FI: Fixed in be294328fb27a8abbe39dd04a13920ed2d413347

Researcher: Verified fix

6.2.3 Emit `BridgeToNonEVMChain` event

Context: AllBridgeFacet.sol#L125

Description: The `_startBridge()` function enables bridging to non-EVM chains but doesn't emit the `BridgeToNonEVMChain` event like other facets. This omission breaks composability.

Recommendation: Consider emitting `BridgeToNonEVMChain` event while bridging to non evm chains.

LI.FI: Fixed in 240e8072a23c400a7ca870b8500ed0fc61d2b0df

Researcher: Verified fix

6.2.4 Inconsistent `InvalidNonEVMReceiver` error

Context: MayanFacet.sol#L43

Description: The `_startBridge()` function in MayanFacet returns `InvalidNonEVMReceiver` error if there exists a mismatch between receiver address in _mayanData and _bridgeData. However this is inconsistent and should use the `InvalidNonEVMReceiver` error from GenericErrors.

Recommendation: Consider streamlining the errors by using `InvalidNonEVMReceiver` from Generic error.

LI.FI: Fixed in af2a8591fa2db65563dd02c353163450bdff79be

Researcher: Verified fix

6.2.5 Validate constructor params

Context: MayanFacet.sol#L57, DeBridgeDlnFacet.sol#L90

Description: The constructor in MayanFacet and DeBridgeDlnFacet should validate the input params, leading to inconsistency with other facets.

Recommendation: Consider validating constructor params in the above mentioned two facets.

LI.FI: Fixed in af2a8591fa2db65563dd02c353163450bdff79be

Researcher: Verified fix

6.2.6 Use NON_EVM_ADDRESS from LiFiData in GasZipFacet

Context: GasZipFacet.sol#L25

Description: Though this file is not under the scope of this audit, it has an inconsistency with the rest of the codebase. Most other facets import NON_EVM_ADDRESS from LiFiData library while this facet uses its own declaration which is redundant.

Recommendation: Consider using NON_EVM_ADDRESS from LiFiData in the above mentioned facet.

LI.FI: Fixed in 8fa11e4658f4fbc70e5154f29e18999cf1382809

Researcher: Verified fix