



LI.FI Security Review

LiFiIntentEscrowFacet.sol (v1.0.1 & v1.1.1)

Security Researcher

Sujith Somraaj (somraajsujith@gmail.com)

Report prepared by: Sujith Somraaj

February 5, 2026

Contents

1	About Researcher	2
2	Disclaimer	2
3	Scope	2
4	Risk classification	2
4.1	Impact	3
4.2	Likelihood	3
4.3	Action required for severity levels	3
5	Executive Summary	3
6	Findings	4
6.1	Informational	4
6.1.1	Inconsistent zero address check methods in <code>LiFiIntentEscrowFacet</code>	4
6.1.2	Missing validation for zero destination chain ID in <code>validateBridgeData()</code> modifier	4

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 Lead 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 security advisor at Superform, a yield aggregator with over \$170M in TVL.

Sujith has experience working with protocols / funds including Coinbase, Uniswap, Layerzero, Edge Capital, Berachain, Optimism, Ondo, 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, Rova, Horizen, 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/LiFiIntentEscrowFacet.sol (v1.0.1 & v1.1.1)

Note on Versioning: This audit report applies to both version 1.0.1 and version 1.1.1 of the LiFiIntentEscrowFacet. The same fix is being applied to two different base versions:

- **v1.0.1:** Based on v1.0.0, deployed immediately as the backend is ready for this version.
- **v1.1.1:** Based on v1.1.0, to be deployed once the backend has been updated to support the newer version.

The change reviewed in this report is identical for both versions, hence a single audit report covers both deployments.

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 1 hours in total, [LI.FI](#) engaged with the [researcher](#) to audit the contracts described in section 3 of this document ("scope"). This is a differential review focussed on the changes from previous version.

In this period of time a total of 2 issues were found.

Project Summary	
Project Name	LI.FI
Repository	lifinance/contracts
Commit	260f8e8
Audit Timeline	February 5, 2025
Methods	Manual Review
Documentation	High
Test Coverage	High

Issues Found	
Critical Risk	0
High Risk	0
Medium Risk	0
Low Risk	0
Gas Optimizations	0
Informational	2
Total Issues	2

6 Findings

6.1 Informational

6.1.1 Inconsistent zero address check methods in `LiFiIntentEscrowFacet`

Context: [LiFiIntentEscrowFacet.sol#L168](#), [LiFiIntentEscrowFacet.sol#L79](#)

Description: The `LiFiIntentEscrowFacet.sol` contract uses two different methods to check for zero addresses:

1. Using `LibUtil.isZeroAddress()`:

```
if (LibUtil.isZeroAddress(_bridgeData.receiver))  
    revert InvalidReceiver();
```

2. Using direct comparison `== address(0)`:

```
if (_lifiIntentData.depositAndRefundAddress == address(0)) {  
    revert InvalidReceiver();  
}
```

Recommendation: Use `LibUtil.isZeroAddress()` consistently throughout the contract for all zero address checks.

LI.FI: Fixed in [f57ec7](#) and [707cb2](#)

Researcher: Verified fix.

6.1.2 Missing validation for zero destination chain ID in `validateBridgeData()` modifier

Context: [LiFiIntentEscrowFacet.sol#L31-L37](#)

Description: The `validateBridgeDataLiFiIntentEscrow()` modifier in `LiFiIntentEscrowFacet.sol` only validates that the receiver is not a zero address and that `minAmount` is non-zero. However, there is no validation for `destinationChainId == 0`.

Recommendation: Add validation for `destinationChainId == 0` in the `validateBridgeDataLiFiIntentEscrow()` modifier

LI.FI: To avoid having facet inconsistencies, we would not implement this check. This check did not exist prior to the change.

Researcher: Acknowledged.