LI.FI Security Review

AcrossFacetV3(v1.1.0)

Independent Review By:

Sujith Somraaj (somraajsujith@gmail.com)

January 6, 2025

Contents

1 About Researcher

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

In addition to working as an external auditor/security researcher with LI.FI, Sujith is a security researcher at Spearbit and Cantina.

Learn more about Sujith on sujithsomraaj.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/AcrossFacetV3.sol(v1.1.0)

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 follow-on review, intended to focus on the hotfix made to the AcrossFacetV3 contracts in lifinance/contracts/pull/911

In this period of time a total of 1 issues were found.

Project Summary	
Project Name	LI.FI
Repository	lifinance/contracts
Commit Hashes	fb4c7a4e02c5fd214d78b8cdc5d4a03fa0b06ab9
Type of Project	
Audit Timeline	January 4, 2024
Methods	Manual Review

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

6 Findings

6.1 Informational

6.1.1 Missing bounds for `outputAmountPercent` parameter in `swapAndStartBridgeTokensViaAcrossV3`

Context: AcrossFacetV3.sol#L111

Description: The **swapAndStartBridgeTokensViaAcrossV3** function lacks bounds checking for **outputAmountPercent**. Values below a reasonable minimum could result in unexpectedly small output amounts, while values above 1e18 would produce inflated outputs.

```
modifiedAcrossData.outputAmount = (_bridgeData.minAmount * _acrossData.outputAmountPercent) / 1e18;
```

Recommendation: Add validation to ensure **outputAmountPercent** is within certain range.

```
require(_acrossData.outputAmountPercent >= MIN_OUTPUT_PERCENT && _acrossData.outputAmountPercent <=
↪  MAX_OUTPUT_PERCENT)
```

LI.FI: I think naming the parameter "percent" is misleading because it could happen that the source and destination chains use different decimals for the two tokens. In this case, the factor can be way higher/lower. This is also why I would not add a limit to the value.

Researcher: Acknowledged. However, some guardrails would be ideal to ensure the value is not zero.